

Salla Alastalo

**IHMISKEISYYDEN JA PALVELUMUOTOILUMETODIEN
HYÖDYNTÄMINEN KETTERÄN OHJELMISTOPROJEKTIN
VAATIMUSMÄÄRITTELYSSÄ**

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
2019

TIIVISTELMÄ

SALLA ALASTALO: Ihmiskeskeisyyden ja palvelumuotoilumetodien hyödyntäminen ketterän ohjelmistoprojektin vaatimusmäärittelyssä

Diplomityö

Tampereen yliopisto

Lokakuu 2019

Tarkastajat: Professori Nina Helander & apulaisprofessori Henri Pirkkalainen

Ketterät menetelmät ovat vakiinnuttaneet paikkaansa ohjelmistokehittäjien keskuudessa. Aihetta on tutkittu paljon, mutta tutkimuskentältä puuttuu kuitenkin oleellisesti konkreettiset keinot siitä, miten projektien vaatimusmäärittelyä voidaan kehittää ihmiskeskeisempään suuntaan. Monet tutkimukset korostavat, että ketterien projektien menestymistä on tehty paljon tutkimuksia ja niiden aikana on huomattu, että vaatimusmäärittelyn onnistuminen vaikuttaa vahvasti myös projektin onnistumiseen. Vaatimusmäärittelyn eteneminen puolestaan riippuu täysin projektin laajuudesta, tarpeista ja asiakkaan omista vaatimuksista.

Tässä tutkimuksessa tutustutaan valmisohjelmiston toiminnallisten vaatimusten määrittämiseen käyttäjäkeskeisesti. Tutkimuksen tavoitteena oli luoda ihmiskeskeisyyteen perustuva viitekehys, jonka lopputuloksena saadaan toteutettavan projektin käyttäjätarinat eli toiminnalliset vaatimukset. Viitekehystä rikastettiin erilaisilla palvelumuotoilumetodeilla, joilla voidaan tukea ihmiskeskeisen suunnittelun periaatteita. Viitekehysten tarkoituksena on mahdollistaa käyttäjätarpeiden määrittäminen siten, että saadaan tietää oikeita asioita oikeilta henkilöiltä tarvittavalla tarkkuudella. Tutkimuksen alussa toteutettiin laaja kirjallisuuskatsaus tutkimuksen aiheesta ja tämän jälkeen kirjallisuuskatsauksen perusteella luotua viitekehystä testattiin empiirisen tutkimuksen avulla. Empiirinen aineisto kerättiin haastatteluiden avulla ja aineistoa analysoitiin temaattisen analyysin keinoin.

Tutkimuksen tuloksena saatiin ketterän ohjelmistoprojektin vaatimusmäärittelylle sopiva viitekehys, joka ottaa huomioon ihmislähtöisen suunnittelun sekä erilaiset palvelumuotoilumetodit. Tutkimus korostaa vaatimusmäärittelyn merkitystä ketterien ohjelmistoprojektien toteutuksessa sekä tarjoaa konkreettisen mallin, jonka avulla toiminnallisia vaatimuksia voidaan mallintaa.

Avainsanat: ihmiskeskeinen suunnittelu, ketterä ohjelmistoprojekti, palvelumuotoilu, vaatimusmäärittely, ketterä vaatimusmäärittely

ABSTRACT

SALLA ALASTALO: Using Human-Centered Design and Service Design Methods in Agile

Software Development Requirements Engineering

Master of Science Thesis

Tampere University

October 2019

Examiner: Professor Nina Helander & Assistant Professor Henri Pirkkalainen

Agile methods have established themselves among software developers. Much research has been carried out into the subject, but the field of research is still lacking in concrete ways of developing requirements specification in more human-centered way. Many studies emphasize that there is lot of studies that handles agile software development methods and during the research it has been noticed that the success of requirements engineering has strong influence on the success of the project. The success of requirements engineering depends on the scope of the project, the needs and the customer's own requirements.

This study examines how to determine the functional requirements of an agile software project with more human-centered approach. The aim of the study is to create a human-centered framework, which will result in user needs or operational requirements for the project to be implemented. The frame was enriched with various service design methods that can support the principles of human-centered design. The purpose of the framework is to enable to determine user needs so that the right things are known with the right accuracy from the right people. At the beginning of the study, an extensive literature review was conducted. After that the reference framework were created based on the literature review and with empirical research the framework was tested. The empirical material was collected through interviews and the material was analyzed through a thematic analysis.

As a result of the study, a suitable framework for the requirements definition of an agile software project was obtained, which considers human-centered design as well as different service design methods. The study highlights the importance of requirement definition in agile software projects and provides a concrete model for recognize and analyze requirements.

Keywords: human-centered design, agile software development, service design, requirements engineering, agile requirements engineering

ALKUSANAT

Tuntuu hurjalta ajatella, että kirjoitan viimeistä työtä koko yliopisto-opintojeni aikana. Viimeiset viisi vuotta ovat kuluneet hurjaa vauhtia ja tuntuu, että juurihan vasta aloitin opiskelut. Samalla kun katsoo taaksepäin voi todeta, että opiskeluvuosiin mahtuu paljon hyviä hetkiä ja unohtumattomia kokemuksia. Näitä en vaihtaisi koskaan pois. Nyt on kuitenkin tullut hetki vaihtaa maisemaa ja siirtyä uusien haasteiden pariin.

Diplomityö toteutettiin kuuden kuukauden mittaisen Trainee-ohjelman aikana. Työ oli laajuudeltaan ja aiheeltaan täysin uudenlainen, mutta todella opettavainen tekijälle. Työn onnistuminen on vaatinut lukuisia hetkiä erilaisten tutkimusten ja artikkelien parissa ja välillä on tuntunut, että työ ei etene ollenkaan. Haluaisinkin kiittää työni ohjaajaa professori Nina Helanderia hyvin tärkeästä tuesta koko diplomityön tekemisen ajan. Kiitos, että kuuntelit epävarmoja ajatuksiani ja osasit auttaa, kun tuntui, että työstä ei tule mitään. Olen iloinen, että yhteistyömme sujui todella hyvin ja kiitos, että osallistuit aktiivisesti työni ohjaamiseen.

Haluan kiittää työpaikkaani ja tutkimuksen toimeksiantajaa mielenkiintoisesta tutkimuksen aiheesta sekä kannustavasta asenteesta tutkimuksen toteutuksen aikana. Kiitos kaikille työntekijöille, jotka osallistuivat työni tekemiseen. Kiitos, että jaoitte omia kokemuksia toteutetuista projekteista avoimesti.

Lopuksi haluan kiittää perhettä ja kaikkia ystäviä, jotka ovat jollain tavalla olleet mukana unohtumattomien muistojen luomisessa sekä tukemassa opiskeluvuosiäni. Kiitos ystävät, että olette olemassa. Erityisen iso kiitos Saijalle, jonka vertaistuki on ollut korvaamaton koko diplomityön tekemisen ajan. Kiitos myös Rikulle kärsivällisestä tuesta sekä opintojen että diplomityön tekemisen ajan. Ilman tätä tukea ja kannustusta en olisi kirjoittamassa näitä alkusanoja.

Tampereella, 22.10.2019

Salla Alastalo

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Tutkimuksen tausta.....	1
1.2	Tutkimuksen tavoitteet, tutkimuskysymykset ja rajaukset.....	2
1.3	Työn rakenne.....	4
2.	TUTKIMUSMETODOLOGIA	7
2.1	Tutkimusasetelma	7
2.2	Kriittinen kirjallisuuskatsaus	12
2.3	Empiirinen tutkimus	13
3.	KETTERÄN OHJELMISTOPROJEKTIN VAATIMUSTEN MÄÄRITTÄMINEN IHMISLÄHTÖISESTI.....	15
3.1	Toiminnallisten vaatimusten määrittely	15
3.1.1	Vaatimusmäärittely ja vaatimusmäärittelyprosessi	18
3.1.2	Ketterä vaatimusmäärittely.....	23
3.1.3	Vaatimusten esillesaanti, kuvaaminen ja priorisointi.....	28
3.2	Käyttäjäkokemus ja ihmislähtöinen suunnittelu	36
4.	PALVELUMUOTOILUN HYÖDYNTÄMINEN KÄYTTÄJÄLÄHTÖISESSÄ SUUNNITTELUSSA	46
4.1	Mitä palvelumuotoilu on?	46
4.2	Vaatimusten mallintaminen palvelumuotoilumetodeja hyödyntäen.....	49
5.	KONSTRUKTIO TEORIASTA	59
5.1	Toiminnallisen vaatimusmäärittelyn prosessimalli.....	59
5.2	Prosessimallin kehittäminen palvelumuotoilumetodeilla	64
6.	EMPIIRISEN TUTKIMUKSEN ANALYYSI JA TULOKSET	67
6.1	Haastattelututkimuksen toteutus	67
6.2	Aineiston analysointi.....	69
6.3	Tutkimuksen tulokset.....	71
6.3.1	Nykytila	71
6.3.2	Esiselvitys.....	75
6.3.3	Tuotteen kehitysjono	78
6.3.4	Vaatimusten hallinta.....	79
6.3.5	Mallin käyttöönotto	80
6.4	Ketterän ohjelmistoprojektin vaatimusmäärittelyn viitekehys	81
7.	YHTEENVETO JA PÄÄTELMÄT.....	87
7.1	Tulosten yhteenveto ja vastaukset tutkimuskysymyksiin	87

7.2 Tutkimuksen arviointi ja uutuusarvo	90
7.3 Tulevaisuuden tutkimuskohteet	92
LÄHTEET	94
LIITTEET	102

LIITE 1: Teorian pohjalta muodostettu viitekehys (Salattu)

LIITE 2: Haastattelurunko

LIITE 3: Teorian ja empirian yhdistämä viitekehys (Salattu)

LIITE 4: Analyysivaiheen rikastettu viitekehys (Salattu)

KUVALUETTELO

Kuva 1.	<i>Tutkimuksen rakenne</i>	5
Kuva 2.	<i>Konstruktivisen tutkimusotteen prosessi (Mukaillen Oyegoken 2011)</i>	10
Kuva 3.	<i>Vaatimusmäärittelyn prosessimalli (Lucia & Qusef 2010)</i>	19
Kuva 4.	<i>Suunnitelmapohjainen kehittäminen muuttumassa arvopohjaiseksi (Schön et al. 2017)</i>	24
Kuva 5.	<i>Ketterä vaatimusmäärittelyn monialaisuus</i>	25
Kuva 6.	<i>Vaatimusten priorisoinnin viitekehys (Lucia & Quesf 2010)</i>	35
Kuva 7.	<i>ISO 9241-210 standardin mukainen ihmislähtöinen suunnitteluprosessi (ISO 2010)</i>	41
Kuva 8.	<i>Ihmiskeskeisen toiminnallisen vaatimusmäärittelyn prosessimalli</i>	60

TAULUKKOLUETTELO

<i>Taulukko 1. Vaatimusten luokittelu eri lähteittäin.....</i>	<i>17</i>
<i>Taulukko 2. Vaatimusmäärittelyn implementointi ketteriin projektimenetelmiin (Lucia & Qusef 2010)</i>	<i>25</i>
<i>Taulukko 3. Ohjeita ketterään vaatimusmäärittelyyn (Lucia & Qusef 2010).....</i>	<i>27</i>
<i>Taulukko 4. Vaatimusten esillesaannin tekniikoita (Nuseibehin & Eastbrook 2000)</i>	<i>28</i>
<i>Taulukko 5. Tekniikat ketterään vaatimusmäärittelyyn (Schön et al. 2016)</i>	<i>29</i>
<i>Taulukko 6. Käyttäjätarinoiden kategorisointi (Girvan & Paul 2017)</i>	<i>30</i>
<i>Taulukko 7. Käyttäjätarinoiden laadulliset ominaisuudet (Girvan & Paulin, 2017; Waken, 2001).....</i>	<i>31</i>
<i>Taulukko 8. Periaatteet käyttäjälähtöiselle suunnittelulle (Gulliksen et al. 2003)</i>	<i>38</i>
<i>Taulukko 9 Käyttökontekstin kuvaamisen osa-alueet (ISO 2010).....</i>	<i>42</i>
<i>Taulukko 10. Ihmislähtöisen suunnitteluprosessin tuotoksia (ISO 2010)</i>	<i>44</i>
<i>Taulukko 11. Palvelumuotoilun periaatteet vuonna 2010 (Stickdorm et al. 2018)</i>	<i>47</i>
<i>Taulukko 12. Palvelumuotoilun muuttuneet periaatteet vuonna 2017 (Stickdorm et al. 2018).....</i>	<i>48</i>
<i>Taulukko 13. Kirjallisuuskatsauksen kautta esille nousseet lähestymistavat</i>	<i>59</i>
<i>Taulukko 14. Ihmiskeskeisen toiminnallisen vaatimusmäärittelyn tunnistettavat tekijät</i>	<i>63</i>
<i>Taulukko 15. Palvelumuotoilumetodit vaatimusmäärittelyprosessissa.....</i>	<i>64</i>
<i>Taulukko 16. Haastateltavien otanta</i>	<i>67</i>

1. JOHDANTO

Tässä luvussa käsitellään diplomityön taustoja sekä tutkimuksen aihetta tarkemmin. Tarkoituksena on selvittää, miksi tutkimusta tehdään, millaisia tärkeitä tutkimuksia alalta jo löytyy sekä määritellään varsinainen tutkimusongelma sekä tutkimuskysymykset. Tämän lisäksi käydään läpi tutkimuksen rajoitteita sekä rakennetta.

1.1 Tutkimuksen tausta

Ketterät menetelmät ovat vakiinnuttaneet paikkaansa ohjelmistokehittäjien keskuudessa. Aiheista on tehty paljon tutkimusta ja niissä on perehdytty mallintamaan erilaisia menetelmiä. Aihealueesta puuttuu kuitenkin oleellisesti konkreettiset keinot siitä, miten projektien vaatimusmäärittelyä voidaan tehostaa ja kehittää erityisesti ihmiskeskeisempään suuntaan. Tarve kehittää toimintatapoja on laajasti tunnistettu, mutta aiheesta ei ole vielä toteutettu kokonaisvaltaisia tutkimuksia.

Tutkimus toteutetaan Digian toimeksiantona. Digia on suomalainen ohjelmisto- ja palveluyritys, joka auttaa asiakkaitaan uudistumaan muuttuvan ympäristön ehdoilla. Digian tarkoituksena on rakentaa maailmaa, jossa digitaalisuus on keskiössä ja toimivaa. (Digia, 2019a) Digia toimittaa ammattitaitoisten ja kokeneiden asiantuntijoiden avulla asiakkailleen uusia palveluita ja toimintatapoja. Yrityksessä työskentelee yli 1200 työntekijää ja yritys on listautunut pörssiyhtiö. Digia toimii muun muassa palveluliiketoiminta, teollisuus ja elintarvike toimialoilla. Yrityksen tavoitteena on luoda arvoa eri sidosryhmille vastuullisella ja kannattavalla liiketoiminnalla. (Digia, 2019b)

Tutkimuksen aihe on muodostunut kohdeyrityksessä esiin tulleista tarpeista. Ohjelmistoprojektien vaatimusmäärittelyä halutaan selkeyttää ja tehdä siitä systemaattinen prosessi, joka tapahtuu ihmiskeskeisiä menetelmiä hyödyntäen. Tällä hetkellä kohdeyrityksessä on hankalaa mallintaa toiminnallisia vaatimuksia niille vaaditulla tarkkuudella. Lähtökohtana on, että toiminnallisia vaatimuksia mallinnetaan prosessikuvasten kautta. Kohdeyrityksessä on kuitenkin tunnistettu tarve, että toiminnalliset vaatimukset tulisi pystyä määrittämään käyttäjäryhmien ja niiden vaatimusten kautta. Tarkoituksena on siirtää prosessilähtöisestä ajattelutavasta ihmiskeskeisempään suuntaan.

Tutkimuksen kannalta on tärkeää ymmärtää, mitä toiminnallisia vaatimuksia tarvitaan, jotta ketterä ohjelmistoprojekti voidaan aloittaa. Vaatimusmäärittelyssä tulee ottaa huomioon ketterä ohjelmistokehitys ja erityisesti erilaiset ketterät menetelmät vaatimusmäärittelylle. Jotta vaatimusmäärittelyn lähestymistapoja voidaan muuttaa ihmiskeskeisempään suuntaan, tulee aiheeseen syventyä kokonaisvaltaisesti.

Tutkimus luo uutuusarvoa yhdistelemällä ketterän ohjelmistoprojektin vaatimusmäärittelyn palvelumuotoilun teoriaan ja metodeihin. Kirjallisuuden perusteella huomataan, että vaatimusmäärittelyä on tutkittu paljon, mutta vielä puuttuu kokonaisvaltainen malli, joka toimisi ketterien ohjelmistoprojektien vaatimusten mallintamisessa. Erityisen mielenkiintoisen tutkimuksesta tekee palvelumuotoilun metodit. Tällä hetkellä on tunnistettu, että ketterän ohjelmistoprojektin vaatimusmäärittelyssä ketterät menetelmät ovat tärkeitä ja niiden avulla saavutetaan paljon tietoa käyttäjien vaatimuksista. Palvelumuotoilumetoodeilla pystytään kuitenkin selvittämään mitä asioita tulee olla tiedossa ja valmiina ennen kuin ketterä ohjelmistoprojekti voidaan aloittaa. Metodit antavat konkreettisesti tietoa siitä, miten asioita voidaan selvittää ja millaisia lopputuloksia on odotettavissa.

Ketterästä ohjelmistoprojektista ja sen menetelmistä on vuosien aikana tehty paljon tutkimusta ja suurin osa tutkimuksista nostaa esille, että vaatimusmäärittely on projektin kannalta hyvin hankala vaihe. Vaatimusmäärittelyn eteneminen riippuu täysin projektin laajuudesta, tarpeista ja asiakkaan omista vaatimuksista. On kuitenkin mahdollista mallintaa tilanteita karkealla tasolla ja siksi tätä tutkimusta lähdetään toteuttamaan.

1.2 Tutkimuksen tavoitteet, tutkimuskysymykset ja rajaukset

Tässä tutkimuksessa tutustutaan valmisohjelmiston toiminnallisten vaatimusten määrittämiseen ihmislähtöisesti sekä määrittelyssä hyödynnettäviin palvelumuotoilumetodeihin. Tutkimuksen tavoitteena on luoda käyttäjäkokemukseen sekä ihmiskeskeisyyteen perustuva viitekehys, jonka lopputuloksena syntyy projektin käyttäjätarinat eli toiminnalliset vaatimukset. Viitekehysten tarkoituksena on mahdollistaa käyttäjätarpeiden määrittäminen siten, että saadaan tietää oikeita asioita oikeilta henkilöiltä tarvittavalla tarkkuudella. Viitekehysten keskiössä on ihmislähtöinen suunnittelu ja miten tämä voidaan ottaa huomioon vaatimusten määrittämisessä.

Viitekehysten tarkoituksena on tukea kommunikointia eri osapuolien välillä sekä varmistaa, että kaikki osapuolet ovat samaa mieltä toiminnallisista vaatimuksista analyysivaiheen loputtua. Palvelumuotoilun avulla pyritään tukemaan käyttäjälähtöistä suunnittelua,

sillä palvelumuotoilumetodien avulla pystytään keräämään käyttäjätarpeita tehokkaasti ja mahdollistetaan standardimuotoisen prosessin toimittaminen riippumatta projektista. Tutkimuksen tavoitteisiin päästään, kun vastataan tutkimukselle asetettuun päätutkimuskysymykseen, joka on:

1. Miten valmisohjelmiston toiminnalliset vaatimukset määritellään käyttäjälähtöisesti?

Koska tutkimuksen päätutkimuskysymys on hyvin laaja, tulee tutkimukselle asettaa apu-tutkimuskysymyksiä, joihin vastaamalla pystytään vastaamaan myös tutkimuksen päätutkimuskysymykseen. Tutkimuksen pääteemoina toimii teorian pohjalta muodostettava viitekehys toiminnallisten vaatimusten määrittämiseen sekä palvelumuotoilun hyödyntäminen vaatimusten määrittämisessä. Tämän takia tutkimuksen alatutkimuskysymyksiä ovat:

1.1 Minkälaisen viitekehysten avulla voidaan kuvata toiminnalliset vaatimukset?

1.2 Miten palvelumuotoilua voidaan hyödyntää toiminnallisten vaatimusten määrittelyssä?

Empiirinen tutkimus pohjautuu teorian pohjalta muodostettuun viitekehykseen ja sen testaamiseen kohdeyrityksessä. Tutkimuksen tuloksena syntyy valmisohjelmiston toiminnallisten vaatimusten käyttäjälähtöiseen määrittämiseen soveltuva viitekehys, joka vastaa tutkimukselle asetettuun päätutkimuskysymykseen.

Tutkimuksen aihe sekä laajuus on määritelty kohdeyrityksen puolesta, jolle diplomityötä tehdään. Tutkimuksen tulee soveltua organisaatioille, joiden investointikyky hankkeeseen on yli 150 000 €. Työn tavoitteena on kuvata käyttäjätarinat sellaisesta näkökulmasta, että niitä voidaan tukea Dynamics 365 -ympäristöllä. Työn ja tuloksien tulee soveltua valmisohjelmistoille ja työkaluna vaatimusten määrittelyssä käytetään Exceliä, jossa on Jiraa vastaava rakenne.

Työssä keskitytään ketterään projektimenetelmään. Ketteryydellä mahdollistetaan nopea reagointi odottamattomiin muutoksiin monipuolisesti ja joustavasti. Ketteryyden avulla saavutetaan monia etuja ja tärkeintä onkin, että sen avulla voidaan toteuttaa muutoksia nopeasti. Muutoksia tehdään laadukkaiden välineiden avulla, jotka ovat yksinkertaisia käyttää. Ympäristön tulee olla joustava, jotta odottamattomiinkin muutoksiin voidaan vastata. (Misra, 2011; Fowler & Highsmith, 2001; Dydå & Dingsoyr, 2008)

Ketterät menetelmät ovat tuoneet ohjelmistokehitykseen ennen näkemättömiä muutoksia. Ketterän ohjelmistokehityksen julistuksen (*Agile Manifesto*) seurauksena kuluneiden vuosien aikana ohjelmistokehitykseen on tuotu paljon erilaisia työkaluja, tekniikoita sekä käytäntöjä, joiden avulla tehostetaan ohjelmistokehitystä. (Chow & Cao, 2008; Misra, 2011). Ketterän ohjelmistokehityksen julistuksessa määritellään: ”Löydämme parempia tapoja tehdä ohjelmistokehitystä kun teemme sitä itse sekä autamme muita siinä. Yhteistyön ja kokemuksen kautta arvostamme:

- **yksilöitä ja vuorovaikutusta** enemmän kuin prosesseja ja työkaluja
- **toimivaa ohjelmistoa** enemmän kuin kattavaa dokumentaatiota
- **asiakasyhteistyötä** enemmän kuin sopimusneuvotteluja
- **muutoksiin reagoimista** enemmän kuin suunnitelman noudattamista”

Sekä oikean, että vasemman puolen tekijöillä on arvoa, mutta vasemmanpuoleisia tulisi korostaa enemmän. (Agile Alliance, 2001; Misra, 2011) Ketterät menetelmät perustuvat joukkoon periaatteita, jotka ohjaavat yleisesti kaikkia erilaisia ketteriä menetelmiä.

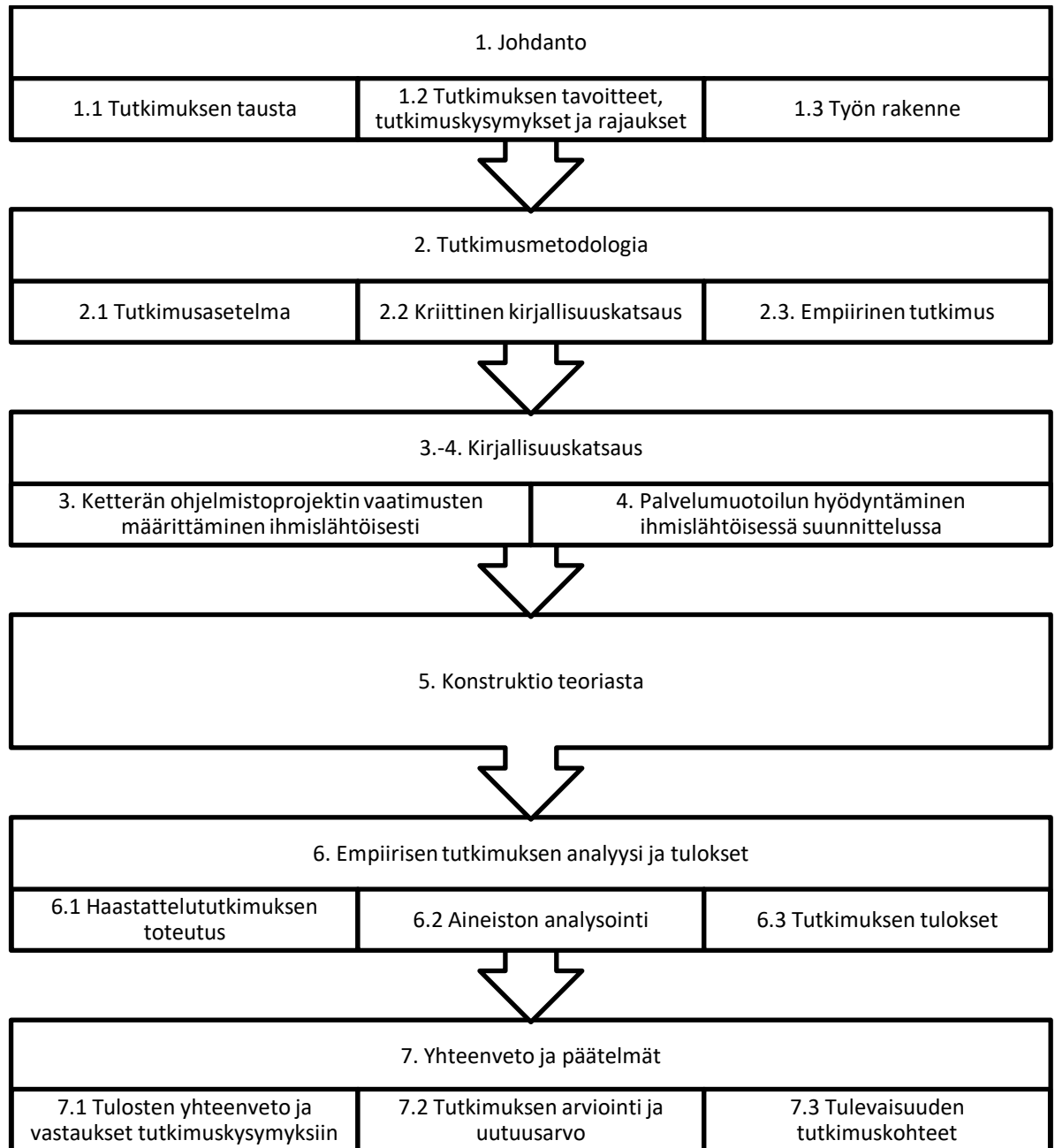
Ketterille ohjelmistoprojekteille on kirjallisuudessa esitelty lukuisia menetelmiä, joista tunnetuimpia ja samalla kirjallisuudessa eniten käsiteltyjä ovat XP ja Scrum. Tässä työssä keskitytään scrum -projektimenetelmään, sillä tutkimuksen kohderyitys hyödyntää ketterien projektien toteutuksessa vastaavanlaista projektitoteutuksen mallia.

1.3 Työn rakenne

Tutkimus on jaettu teoriaosuuteen sekä empiiriseen tutkimukseen. Ennen teoriaosuutta käydään johdannossa läpi tutkimuksen tausta, tavoitteet, rajoitteet sekä rakenne. Tämän lisäksi toisessa luvussa käydään läpi tutkimuksen tutkimusmetodologia. Tutkimusmetodologian avulla selvennetään sitä, millainen tutkimusasetelma tutkimukselle on valittu ja millaisia metodeja työssä käytetään. Tämän lisäksi käydään läpi, miten kirjallisuuskatsaus on toteutettu ja miten empiirinen tutkimus liittyy tutkimuksen kokonaisuuteen. Kuvassa 1 on esitetty tutkimuksen rakenne.

Teoriaosuus on jaettu kahteen osaan, joista ensimmäinen osuus käsittelee ketterän ohjelmistoprojektin toiminnallisten vaatimusten määrittämistä. Tarkoituksena on toteuttaa kirjallisuuskatsaus valituista aiheista. Teoriaosuuden toisessa osassa käsitellään palvelumuotoilua sekä palvelumuotoilun metodeja, joiden avulla voidaan tehostaa toiminnal-

listen vaatimusten määrittelyä. Teoriaosuuden lopputuloksena saadaan viitekehys ketterän ohjelmistoprojektin toiminnallisten vaatimusten määrittämiseen sekä käyttäjätarinoiden mallintamiseen. Tämä viitekehys esitellään luvussa 5, johon on koottu teorian pohjalta muodostettu konstruktio.



Kuva 1. *Tutkimuksen rakenne*

Työn empiirinen tutkimusosuus keskittyy teorian pohjalta luodun viitekehysten testaamiseen sekä arviointiin. Luvussa kuusi keskitytään työn empiriaan ja testataan mallin toimivuutta kohdeyrityksessä. Tarkoituksena on saada palautetta mallin toimivuudesta sekä konkreettisia kehitysehdotuksia. Tämän testaamisen avulla mallia parannetaan,

jotta siitä saadaan kokonaisuudessaan toimiva menetelmä. Näiden asioiden lisäksi viimeisessä luvussa käsitellään työn yhteenvetoa ja vastataan tutkimukselle asetettuihin tutkimuskysymyksiin. Yhteenvedossa arvioidaan myös tutkimuksen tuloksia ja uutuusarvoa sekä pohditaan mahdollisia tulevaisuuden tutkimuskohteita.

2. TUTKIMUSMETODOLOGIA

Tässä luvussa esitellään tutkimusasetelma sekä käydään läpi kriittisen kirjallisuuskatsauksen ja empiirisen tutkimuksen toteutustavat. Tarkoituksena on tutustua tutkimuksen filosofiseen lähestymistapaan sekä esitellään metodit, joita tutkimuksessa käytetään. Kriittisen kirjallisuuskatsauksen kappaleessa esitellään tutkimuksen tiedonhakuprosessi, joka antaa lukijalle kuvan siitä, miten tutkimuksen materiaalit on kerätty. Tämän lisäksi käydään läpi empiirinen tutkimuksen toteutus.

2.1 Tutkimusasetelma

Tutkimusta tehtäessä on aluksi valittava menetelmällinen lähestymistapa sekä tiedostettava lähestymistavan taustaoletukset sekä rajoitteet. Tutkimuksille on yleisesti olemassa kaksi erilaista lähestymistavan pääsuuntausta, joita ovat deduktio ja induktio. Näiden kahden lähestymistavan lisäksi on olemassa myös näiden yhdistelmä abduktiivinen järjely. (Saunders et al. 2009, ss. 61-63) Deduktiivisessa järjelyssä tutkimuksessa lähdetään liikkeelle teoriasta, jonka jälkeen syvennyttään empiiriselle tasolle. Deduktiivisessa järjelyssä päättely pohjautuu oletettuun teoriaan, jota testataan. Lopulta teoria joko vahvistuu tai ei. (Saunders et al. 2009, ss. 124-125) Induktiivinen tutkimus puolestaan lähtee liikkeelle empiirisestä tutkimuksesta ja täydentää sillä teoriaa. Induktiivinen järjely perustuu havaintojen tekemiseen. Näistä havainnoista pyritään tunnistamaan toistuvia tekijöitä ja näiden perusteella luodaan uusia teorioita. (Saunders et al. 2009, ss. 125-126; Eriksson & Kovalainen, 2015, ss. 21-22)

Tämän tutkimuksen suhde teoriaan ei ole suoraviivaisesti deduktiivinen tai induktiivinen vaan ennemminkin yhdistelmä deduktiivisesta ja induktiivisesta lähestymistavasta. Tutkimus sisältää teorian ja käytännön vuoropuhelua ja siksi tutkimus noudattelee abduktiivista järjelyä. (Eriksson & Kovalainen, 2015, ss. 22-25; Saunders et al. 2009, ss. 125-126) Abduktiolle on tyypillistä, että se perustuu empiriaan, mutta teoria toimii tutkimuksen aikana hyvänä luovuuden lähteenä ja sitä hyödynnetään työssä monessa tilanteessa. Abduktiivisen lähestymistavan avulla pystytään tekemään laajempia päätelmiä aiheesta. (Walton, 2014, ss. 3–7) Tässä tutkimuksessa abduktiivisuus tulee esille siinä, että työssä muodostetaan kirjallisuuden pohjalta malli, jonka pohjalta toteutetaan haastattelut sekä testataan mallin toimivuutta. Haastatteluiden pohjalta pystytään arvioimaan mallin toimivuutta eli hypoteesia. Haastatteluiden avulla laajennetaan ymmärrystä aiheesta ja mallin

toimivuudesta sekä niiden pohjalta voidaan muodostaa tarpeen tullen uusi hypoteesi eli viitekehys toiminnallisten vaatimusten määrittämiseen.

Tutkimuksen lähestymistavan lisäksi tulee tutkimukselle valita tutkimusstrategia. Työn tarkoituksena on kehittää viitekehys valmisohjelmiston toiminnallisten vaatimusten määrittämiseen sekä käyttäjätarinoiden kuvaamiseen. Työssä hyödynnetään sosiaalista konstruktionismia ja tutkimuksen tutkimusstrategiana on tapaustutkimus, jossa tarkastellaan ja analysoidaan intensiivisesti yhtä tai useampaa valittua tapausta. Tapaustutkimuksen tapauksella tarkoitetaan esimerkkiä tai ilmentymää ilmiöstä, jota tutkitaan syvällisesti. (Saunders et al. 2009, ss. 145-146) Tapauksen tarkastelu on hyvin tilannekohtaista ja tässä tutkimuksessa tarkastelun kohteena käytetään kohdeyritystä, jolle tutkimusta toteutetaan.

Tapaustutkimukselle selviä vahvuuksia ovat tulosten olennaisuus erityisesti johtamisen näkökulmasta. Tapaustutkimuksen avulla pystytään lisäämään ymmärrystä sekä tarjoamaan mahdollisuuden tarkastella asioita hyvin syvällisellä tasolla. Tapaustutkimus vastaa myös kysymyksiin, miten ja miksi. (Yin, 2013; Eriksson & Kovalainen, 2015, s. 116) Haasteellista puolestaan on tapaustutkimuksen työmäärän paljous ja tutkimus vaatii usein useamman menetelmän yhtäaikaista käyttöä sekä vuoropuhelua menetelmien välillä (Saunders et al. 2009, s. 145).

Tapaustutkimuksen lisäksi tutkimuksessa hyödynnetään konstruktiivista tutkimusotetta. Konstruktiiviselle tutkimusotteelle on tyypillistä, että se on soveltavaa empiiristä tutkimusta. Uuden tiedon avulla tähdätään johonkin tavoitteeseen kuten ongelmanratkaisuun tai innovaatioon. Konstruktiivinen tutkimusote ottaa kantaa siihen, miten pitäisi toimia ja on luonteeltaan normatiivista. Tutkimuksessa lähdetään liikkeelle nykytilasta, johon halutaan saada aikaan konstruktio eli jokin muutos. Konstruktiolla tarkoitetaan yleisesti viitekehystä tai käsitteellistä rakennelmaa, joka pitää sisällään ratkaisun ennalta määritellyn ongelmaan. (Kasanen et al. 1993)

Konstruktiivista tutkimusotetta käytetään ongelman määrittämisessä sekä ratkaisemisessa. Sen avulla voidaan parantaa jo olemassa olevaa järjestelmää tai suorituskykyä. Konstruktiivista tutkimusta voidaan Oyegoken (2011) mukaan luonnehtia soveltavaksi tutkimukseksi, joka usein johtaa uusiin tietoihin normatiivisten sovellusten muodossa. Oyegoken (2011) mukaan analyyttinen mallinrakennus tuottaa elegantin todistetun ratkaisun, joka toimii periaatteessa, mutta jonka todellinen käytännöllinen riittävyys on yleensä epäselvää. Konstruktiivinen tutkimusote perustuu konstruktion rakentamiseen.

Kasanen et al. (1993) mukaan konstruktiolla tarkoitetaan ratkaisun tuottamista hyvin tarkasti määritettyyn ongelmaan. Usein tämä ratkaisu tuotetaan muutosprosessin avulla, jonka tuotoksena syntyy jotain, täysin erilaista, jollaista ei aikaisemmin ole ollut.

Kasanen et al. (1993) toteaa, että kaikki ongelmanratkaisutilanteet eivät sovellu konstruktiiviseen tutkimukseen. Konstruktiivisessa lähestymistavassa sidotaan ongelma ja ratkaisu yhteen teoreettisten tietojen kanssa. Konstruktiivisen lähestymistavan ytimessä on innovaatio tai konstruktion suunnitelma. Ratkaisun uutuus ja todellinen toiminta tulee osoittaa tutkimuksessa. Oyegoken (2011) mukaan konstruktiivinen lähestymistapa edellyttää innovointia, luovuutta sekä läpinäkyvyyttä.

Konstruktiiviselle tutkimukselle on tyypillistä, että koko prosessin ajan pidetään päämäärä ja tavoitteet selkeänä. Pyritään alussa selventämään sitä, mitä vaikutuksia konstruktiolta odotetaan. Yhtä tärkeää on myös ratkaisun toimivuuden testaaminen. (Lukka, 2001, Ojasalo et al. 2009, ss. 67-68) Tutkimusotteelle on tunnistettavissa prosessisykli, joka on esitetty kuvassa 2. Prosessisykli sisältää kuusi erilaista vaihetta, joita noudattamalla pystytään toteuttamaan konstruktiivinen tutkimus. Ensimmäinen vaihe on käytännön ongelman määrittäminen. Tässä vaiheessa on tarkoituksena löytää käytännön kannalta merkityksellinen ongelma, jolla on tutkimuspotentiaalia. (Ojasalo et al. 2009, ss. 67-68)

Tämän tutkimuksen tutkimusongelma on muodostunut kohdeyrityksen tarpeesta. Heillä on pidemmän aikaa tiedostettu muuttuva tilanne ketterien ohjelmistoprojektien keskuudessa. Nykyään asioita halutaan nähdä enemmän ihmiskeskeisenä eikä prosessikeskeisenä. Tällä hetkellä ongelmana on, että ei ole mitään systemaattista tapaa sekä löytää, että mallintaa toiminnallisia vaatimuksia niille vaaditulla tasolla. Tiedostetaan, että suurissa projekteissa vaatimukset tulee määrittää tarkasti, mutta mitään menetelmää tai työkalua tähän ei ole. Ongelmana on myös se, että ei tiedetä, kuinka tarkasti vaatimukset tulee määrittää ja miten ne saadaan sellaiseen muotoon, jonka jokainen projektiin osallistuva ymmärtää. Tutkimuksen tarkoituksena on vastata näihin ongelmiin sekä ottaa kantaa siihen, miten toiminnalliset vaatimukset pystytään mallintamaan mahdollisimman tehokkaasti palvelumuotoilumetodeja hyödyntäen.



Kuva 2. *Konstruktiiivisen tutkimusotteen prosessi (Mukaillen Oyegoken 2011)*

Prosessin toinen vaihe on käytännön ja teoreettisen tiedon hankinta valitusta aiheesta. Tässä prosessivaiheessa luodaan yleinen sekä kokonaisvaltainen käsitys tutkittavasta aiheesta. (Oyegoken, 2011) Tärkeää on tutkia mitä tähän mennessä kirjallisuudessa on kirjoitettu aiheesta. Jotta tutkittavasta asiasta saadaan kattava käsitys, tulee tutkimuslinjaa laajentaa edelleen toisiinsa liittyviin tieteenaloihin ja käytäntöihin sekä vertailtava aiheita keskenään muodostaen kattavan synteesin aiheesta. Tässä vaiheessa tulee tarkastella perusteellisesti useita tutkimuksia erityisesti sellaisia, joissa käsitellään innovatiivisten ongelmien ratkaisemista. (Ojasalo et al. 2009, ss. 67-68; Oyegoken, 2011)

Tässä tutkimuksessa sovelletaan prosessin toista vaihetta hyvin suoraviivaisesti. Teoriaosuutta lähdetään työstämään toiminnallisten vaatimusten ja vaatimusmäärittelyn kautta. Ensin selvitetään, mitä näillä kahdella termillä tarkoitetaan ja miten vaatimusmäärittelyä voidaan mallintaa prosessina. Tutkimuksen edetessä ja tiedon karttuessa mukaan nostetaan myös tekijöitä, joita voidaan aikaisempien tutkimusten perusteella pitää relevantteina. Teoriaa täydennetään palvelumuotoilun määritelmällä sekä metodeilla. Kun ensin on saatu kokonaisvaltainen kuva siitä, millainen prosessi toiminnallisten vaatimusten määrittäminen on, voidaan siihen soveltaa palvelumuotoilun metodeja kirjallisuuden avulla.

Prosessin kolmannessa vaiheessa tarkoituksena on innovatiivisten teoreettisten ratkaisujen ja käytännön konstruktoiden tuottaminen (Ojasalo et al. 2009, s. 67). Konstruktiiivisessa lähestymistavassa konstruktio perustuu kontekstuaaliseen kirjallisuuden tarkasteluun ja ongelman käytännönläheisyyteen. Laaja-alaisten kirjallisuuskatsauksien tulisi auttaa tutkijaa saamaan perusteellinen esiselvitys kohdennetusta ilmiöstä. Tässä vaiheessa luodaan konstruktio, jota testataan myöhemmissä vaiheissa prosessia. (Toikko & Rantanen, 2009, ss. 19-23; Ojasalo et al. 2009, ss. 67-68; Oyegoken, 2011)

Tutkimuksen konstruktiossa kirjallisuuskatsauksen pohjalta muodostetaan malli toiminnallisten vaatimusten määrittämiseen. Malli ottaa kantaa siihen, millainen prosessi vaatimusmäärittely ketterissä ohjelmistoprojekteissa on ja miten vaatimukset saadaan selville sekä miten niitä tulee mallintaa. Mallin muodostamisen jälkeen jatketaan konstruktion muodostamista ja rikastetaan mallia erilaisten palvelumuotoilumetodien avulla. Palvelumuotoilumetodit antavat konkreettista tietoa siitä, millaisten toimintojen avulla vaatimuksia voidaan saada selville ja millaisia tuloksia niistä saadaan.

Prosessisyklin neljännessä vaiheessa testataan ja todistetaan uuden konstruktion eli ratkaisun toimivuus (Ojasalo et al. 2009, s. 68). Konstruktiivisessa tutkimuksessa voidaan käyttää sekä kovia, että pehmeitä paradigmoja, joiden avulla osoitetaan uuden ratkaisun toimivuus. Pollackin (2007) mukaan koviin paradigmoihin perustuva käytäntö pyrkii korostamaan tehokasta, asiantuntijoiden johtamaa toimitusta, valvomaan ennalta määritellyjä tavoitteita ja kiinnostusta taustaratkaisuun. Pehmeät paradigmat puolestaan korostavat oppimista, osallistumista, hankkeiden helpottamista ja tyypillisesti kiinnostusta sosiaaliseen prosessiin. Sopivin tapa testata ja parantaa rakennetta tehdään pilottitapaustutkimuksessa.

Tutkimuksen empiria keskittyy testaamaan ja todistamaan konstruktion avulla muodostettua mallia. Tarkoituksena on haastatella kohdeyrityksen henkilöitä, joilla on kokemusta toteutetuista analyysivaiheista. Haastatteluiden avulla selvennetään millaisena haastateltavat näkevät nykytilanteen ja miten konstruktion avulla muodostettu malli voisi toimia yrityksen analyysivaiheissa. Haastatteluiden aikana käydään läpi analyysivaiheen tavoitteita ja peilataan niitä muodostettuun malliin ja sen toimivuuteen. Empiirisen tutkimuksen lopputuloksena saadaan testattu ja validoitu viitekehys, joka toimii tutkimukselle asetettujen tavoitteiden puitteissa.

Viidennessä vaiheessa identifioidaan ja analysoidaan teoreettinen osio. Tutkimuksessa otetaan lähestymistavan puitteissa huomioon vakiintuneet tutkimusmenettelyt sekä on dokumentoitava toiminnalliset mitattavat tiedot yksityiskohtaisesti. (Oyegoken 2011) Tämän avulla luodaan luotettavuutta, sillä jokainen vaihe on mahdollista tarkistaa. Jotta konstruktiivisessa tutkimuksessa voidaan määrittellä tietovaje ja tutkimusongelma, tulee kriittinen ongelma määrittää ja tutkia perusteellisesti. Yhdistettyjen teoreettisten ja empiiristen perusteiden pohjalta uusi ratkaisu on suunniteltava ja validoitava esimerkiksi pilotoinnin, teoreettisen validoinnin, empiirisen validoinnin ja analogisen validoinnin

avulla. (Ojasalo et al. 2009, ss. 67-68; Toikko & Rantanen 2009, ss. 19-23) Konstruktiivinen tutkimusote vaatii, että konstruktion tulee lisätä tietämystä ja teoreettiset kontribuutiot tulisi perustella. Myös uutuus ja soveltamisalat tulisi mainita selvästi. (Oyegoken, 2011)

Prosessin viimeisessä kuudennessa vaiheessa viimeistellään konstruktio ja määritetään käytännön soveltamismahdollisuudet. Konstruktiivinen tutkimusote edellyttää, että tutkimuksen kumulatiivinen tietämys on määriteltävä ja lisätutkimusalueet tulee ottaa huomioon. (Oyegoken, 2011)

2.2 Kriittinen kirjallisuuskatsaus

Kriittisen kirjallisuuskatsauksen merkitys tutkimukselle on hyvin suuri. Saunders et al. (2009, ss.58-59) mukaan kriittisen kirjallisuuskatsauksen päätarkoitus on auttaa tutkimuksen tekijää kehittämään hyvän käsityksen siitä, mitä aihetta ollaan tutkimassa ja millaisia tutkimuksia aiheen tiimoilta on aikaisemmin tehty. Samalla kirjallisuuskatsauksen avulla voidaan tukea tutkimusideoiden luontia sekä ymmärtää millaista lähestymistapaa tutkimuksessa kannattaa hyödyntää. Saunders et al. (2009, s. 61) nostaa myös esille sen, että tarkoituksena ei ole tuottaa yhteenvetoa kaikista tutkimuksista, mitä aihealueesta on aikaisemmin tehty, mutta olisi tärkeää löytää ne kaikista oleellimmat ja relevantteimmat tutkimukset aihealueesta.

Koska kyseessä on konstruktiivinen tutkimus, luodaan kokonaisvaltainen kuva tutkittavasta aiheesta kriittisen kirjallisuuskatsauksen avulla. Kirjallisuuskatsauksen suorittamisessa hyödynnettiin Saunders et al. (2009, ss. 58-62) kirjallisuuskatsauksen ohjeita. Saunders korostaa, että vaikka ajattele, että omaat hyvän tietämyksen tutkittavasta aiheesta, tulee kirjallisuuskatsaus toteuttaa tutkimuksessa ja sen merkitys on kriittinen. Tärkeää on kirjallisuuskatsauksen avulla vahvistaa ja tuoda esille, mitä tutkimuksia on jo julkaistu valitusta aiheesta sekä yrittää tunnistaa, mitkä tutkimusalueet voisivat olla tällä hetkellä tarkastelun alla. Vaikka suurimmassa osassa tutkimuksista kirjallisuuskatsaus tehdään hyvin varhaisessa vaiheessa Saunders et al. (2009, s. 60) korostavat sitä, että lähteitä tulisi etsiä koko tutkimuksen toteutuksen ajan, sillä uusia näkökulmia ja tarpeita paljastuu aina tutkimuksen edetessä.

Onnistuneen kirjallisuuskatsauksen lähtökohtana on tutkimuskysymysten tarkka muodostaminen ja niiden pohjalta tärkeiden avainsanojen määrittäminen. Näiden avainsanojen pohjalta tehdään ensimmäisiä hakuja tutkittaviin aineistoihin. Yksittäisten, ensimmäisten hakujen jälkeen on mahdollista määrittää parametreja vielä tarkemmin. Kriittinen kirjallisuuskatsaus tarjoaa hyvän pohjan koko tutkimukselle. Tärkein tehtävä on auttaa tutkijaa ymmärtämään jo aikaisemmin tehtyjä tutkimuksia sekä tunnistamaan trendejä, joita aiheeseen liittyy.

Tämän tutkimuksen kirjallisuuskatsaus koostuu suurimmaksi osaksi kirjoista, artikkeleista ja konferenssijulkaisuista. Materiaalia on rajattu suomen- ja englanninkielisiin julkaisuihin ja hakurajauksena myös ”tieteellinen ja vertaisarvioitu”, jotta hakutulokset täyttävät tarvittavat laadulliset kriteerit. Hakukriteerit täyttäviä materiaaleja käytiin läpi myös tietyin periaattein. Ensinnäkin tarkasteltiin artikkelin otsikkoa ja tämän perusteella päätettiin, onko artikkeli relevantti tutkimukselle. Jos artikkeli hyväksyttiin otsikon perusteella tai tarvittiin enemmän tietoa artikkelin sisällöstä, perehdyttiin artikkelin tiivistelmään sekä arvioitiin sen sopivuutta tutkimukselle. Artikkelit, jotka hyväksyttiin tiivistelmän perusteella, luettiin kokonaan ja artikkelin relevanttiutta arvioitiin luetun sisällön perusteella.

2.3 Empiirinen tutkimus

Kriittisen kirjallisuuskatsauksen tuloksena tutkimuksessa kehitetään malli, jolla voidaan mallintaa käyttäjien toiminnallisia vaatimuksia. Koska malli muodostetaan teorian pohjalta valittuun aiheeseen, tulee sitä testata tutkimuksen seuraavassa vaiheessa. Tutkimuksen empiria keskittyy viitekehyksen testaamiseen kohdeyrityksessä.

Empiriassa tapahtuva tiedonkeruu suoritetaan kvalitatiivisten puolistrukturoitujen haastatteluiden avulla. Empirian avulla pyritään ymmärtämään haastatteluiden avulla henkilöiden mielipiteitä ja asenteita ja siksi puolistrukturoitu haastattelutekniikka sopii tämän tutkimuksen empiriseen osuuteen. (Saunders et al. 2009, ss. 323-324) Tutkimuksen haastattelut ovat teemahaastatteluita, jotka ovat hyvin käytettyjä tutkimusmenetelmiä, joissa hyödynnetään keskustelumutoista haastattelumenetelmää, joka sopii laadulliseen tutkimukseen. Teemahaastattelulle on tyypillistä, että haastattelun kysymykset ovat kohdistettu valittuihin teemoihin. Kaikkien kysymysten ei kuitenkaan tarvitse olla täydellisesti muotoiltuja. Tämä antaa enemmän tilaa keskustelulle, mutta samalla vaatii haastattelijalta hyvää valmistautumista. Haastattelijalla tulee olla tietoa aiheesta etukäteen, jotta oikeita kysymyksiä pystytään kysymään haastattelun aikana. (Hirsijärvi & Hurme, 2011, ss. 46-48; Saunders et al. 2009, s. 324) Kun haastatteluita suunnitellaan, tulee

miettiä kysymysten asettelua ja merkitystä. Haastateltavien määrä tulisi Malterund et al. (2016) mukaan valita aina tapauksen perusteella. Näiden lisäksi haastatteluita ennen tulisi suunnitella se, minkä tyyppinen haastattelu tehdään ja miten saatua tietoa analysoidaan ja käsitellään (Qu & Dumay, 2011).

Haastatteluiden haastateltavat henkilöt on valittu harkintana pohjautuvaa otantaa hyödyntäen. Harkintaan pohjautuvassa menetelmässä haastateltavat valitaan rajaavien kriteerien perusteella. Rajaavat kriteerit mahdollistavat sen, että haastatteluiden otanta koostuu parhaimmista haastateltavista ja haastatteluiden avulla saadaan hyvin todennäköisesti selkeitä vastauksia tutkimuskysymyksiin. (Saunders et al. 2009, s. 240) Rajaavina kriteereinä toimii henkilön työtehtävä ja tiimi, johon kuuluu. Tämän lisäksi haastateltavia valittaessa painotettiin haastateltavan kokemusta omissa työtehtävissä, sillä haluttiin saada mahdollisimman kokonaisvaltaiset tulokset hyödyntämällä vain muutamaa haastattelua.

3. KETTERÄN OHJELMISTOPROJEKTIN VAATIMUSTEN MÄÄRITTÄMINEN IHMISLÄHTÖISESTI

Tässä luvussa keskitytään ketterän ohjelmistoprojektin toiminnallisten vaatimusten määrittämiseen sekä tunnistetaan kirjallisuuden avulla vaatimusmäärittelylle oma prosessimalli. Vaatimusten osalta käydään läpi, miten vaatimuksia voidaan luokitella, analysoida ja priorisoida sekä miten vaatimuksia voidaan kuvata mahdollisimman tehokkaasti käyttäjätarinoiden muodossa. Tämän lisäksi tutustutaan ihmislähtöiseen suunnitteluun ja miten sitä voidaan hyödyntää toiminnallisten vaatimusten määrittämisessä.

3.1 Toiminnallisten vaatimusten määrittely

Yleisesti vaatimuksilla tarkoitetaan ominaisuuksia, jotka kuvaavat hankinnan kohteena olevaa tietojärjestelmää. Vaatimukset toimivat suunnittelun, toteutuksen sekä testaamisen pohjana järjestelmälle. (Young, 2004) Hull et al. (2005) mukaan vaatimukset kuvaavat järjestelmän lähtökohtia. Vaatimukset tulee ilmaista yleiskielellä ja niiden tulee olla ymmärrettäviä, jotta ne kuvaavat sitä, mitä käyttäjät, asiakkaat, kehittäjät ja muut yhteisöryhmänsä tarvitsevat järjestelmältä ja mitä järjestelmältä vaaditaan, jotta se pystyy vastaamaan näihin tarpeisiin. Kotonya ja Sommerville (2002, ss. 7-8) kuvaavat vaatimukset ohjelmiston avulla ratkaistavien ongelmien kuvaukseksi. Vaatimusten avulla kuvataan vaaditut ohjelmiston toiminnallisuudet, ominaisuudet sekä suunnittelun ja toteutuksen rajoitteet. Kotongan ja Sommervillen (2002, ss. 7-8) mukaan vaatimus koostuu käyttäjän kokemasta toiminnallisuudesta, yleisistä järjestelmän ominaisuuksista, järjestelmien reunaehdoista sekä järjestelmän kehittämisen reunaehdoista.

Käyttäjätarpeella tarkoitetaan sitä, mitä käyttäjä tarvitsee järjestelmältä (Lucia & Qusef, 2010). Käyttäjätarpeille on tyypillistä se, että ne voivat olla hyvin ylätasen määritelmiä eikä niiden tarvitse olla niin formaaleja kuin käyttäjävaatimusten. On hyvä ymmärtää näiden kahden termin erot. Käyttäjävaatimukset ovat pilkottuja osia käyttäjätarpeista, jotka ovat järjestelmän ominaisuuksia, jotka vastaavat tiettyyn käyttäjätarpeeseen. Koska käyttäjätarpeet voivat olla karkeita ylätasen määritelmiä, jättävät ne paljon tulkinnanvaraa. Jotta näitä käyttäjätarpeita voidaan ymmärtää paremmin, pilkotaan ne palasiksi käyttäjävaatimusten muotoon. (Pohl & Rupp, 2011, ss. 3-5; Kotonya & Sommerville, 2002, ss. 7-8)

IEEE standardin 610.12-1990 (IEEE, 1990) mukaan vaatimus on:

1. Edellytys tai kyvykkyys, jonka avulla käyttäjä voi ratkaista ongelman tai saavuttaa tavoitteen
2. Edellytys tai kyvykkyys, jonka järjestelmän tai järjestelmäkomponentin on saavutettava tai täytettävä, jotta sopimus, standardi, määritys tai muu formaali dokumentin ehto täyttyy
3. Dokumentoitu esitys edellisissä kohdissa (1 tai 2) kuvatuista edellytyksistä tai kyvykkyyksistä

Yleisesti voidaan ajatella, että vaatimukset ovat järjestelmän kehittämisen rajoitteita ja implementoinnin ohjeistusta. Vaatimusten avulla kuvataan se, miten järjestelmän tulee toimia. Tulee kuitenkin muistaa, että vaatimusten avulla ei oteta kantaa tekniseen toteutukseen. Ensin määritetään vaatimukset ja vasta tämän jälkeen siirrytään suunnittelemaan teknistä toteutusta. (Hull et al. 2005; Kotonya & Sommerville, 2002, ss. 7-8, s. 28; Robertson & Robertson, 2011, s. 2-4)

Vaatimusten luokittelu

Vaatimuksia voidaan luokitella järjestelmän toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. JUHTA (2009) määrittelee myös toimintälähtöiset vaatimukset sekä käyttäjävaatimukset omaksi ryhmäksi. Toimintälähtöiset vaatimukset tulee johtaa käyttäjävaatimuksista. Vastaavasti toiminnalliset ja ei-toiminnalliset vaatimukset pohjautuvat käyttäjävaatimuksiin. (JUHTA, 2009) Myös Lamsweerden (2009) sekä Kotonya ja Sommerville (2002) jakavat vaatimukset toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Lamsweerden (2009) mukaan järjestelmien toiminnalliset vaatimukset kuvaavat järjestelmän toimintoja ja ei-toiminnalliset vaatimukset keskittyvät järjestelmän rajoituksiin. Kotonya ja Sommerville (2002) puolestaan esittävät, että ei-toiminnalliset vaatimukset kuvaavat järjestelmien yleisiä ominaisuuksia sekä piirteitä ja asettavat järjestelmälle ehdot, miten käyttäjän vaatimukset tulee toteuttaa. Myös Haikala ja Mikkonen (2011) jakavat vaatimukset toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Tämän lisäksi he nostavat kolmanneksi vaatimusten luokaksi reunaehdot tai rajoitteet.

Pohl ja Rupp (2011) sekä Robertson ja Robertson (2011) mukaan toiminnalliset vaatimukset kuvaavat toiminnallisuuksia, jotka järjestelmän tulee täyttää ja siksi toiminnalliset vaatimukset voidaan heidän mukaansa jakaa toiminnallisuuksien vaatimuksiin, käyttäytymisen vaatimuksiin sekä tietovaatimuksiin. Ei-toiminnallisilla vaatimuksilla kuvataan

järjestelmän tärkeimpiä haluttuja ominaisuuksia kuten suorituskykyä, saavutettavuutta, luotettavuutta ja skaalautuvuutta.

Hull et al. (2005, ss. 4-5) mukaan vaatimukset voidaan jakaa käyttäjävaatimuksiin, järjestelmävaatimuksiin ja arkkitehtuurisuunnitteluun. Järjestelmävaatimuksilla kuvataan abstraktilla tasolla sitä, kuinka järjestelmä tulee vastaamaan käyttäjävaatimuksiin. Käyttäjävaatimusten avulla kuvataan se, mitä käyttäjät sekä sidosryhmät haluavat järjestelmän avulla saavuttaa. Arkkitehtuurisuunnittelulla tuodaan esille se, kuinka jokin ratkaisu toteuttaa järjestelmävaatimukset. (Hull et al. 2005, s. 16)

Taulukko 1. *Vaatimusten luokittelu eri lähteittäin*

Lähde	Luokittelu
<i>JUHTA (2009)</i>	Toimintälähtöiset vaatimukset Käyttäjävaatimukset Toiminnalliset vaatimukset Ei-toiminnalliset vaatimukset
<i>Lamsweerde (2009)</i>	Toiminnalliset vaatimukset Ei-toiminnalliset vaatimukset
<i>Kotonya ja Sommerville (2003)</i>	Toiminnalliset vaatimukset Ei-toiminnalliset vaatimukset
<i>Haikala ja Mikkonen (2011)</i>	Toiminnalliset vaatimukset Ei-toiminnalliset vaatimukset Reunaehdot ja rajoitteet
<i>Hull et al. (2005)</i>	Käyttäjävaatimukset Järjestelmävaatimukset Arkkitehtuurisuunnittelu

Taulukkoon 1 on koottu kaikki vaatimusten luokittelut eri lähteiden mukaan. Taulukosta huomataan, että vaatimuksia voidaan luokitella aina hieman eri tavalla riippuen siitä, kuka tarkastelua tekee. Yleistä on kuitenkin, että vaatimukset jaetaan toiminnallisiin ja

ei-toiminnallisiin vaatimuksiin. Näiden kahden luokittelun välillä suurin ero tulee siitä, että toiminnalliset vaatimukset keskittyvät nimensä mukaan kuvaamaan järjestelmän toimintoja, kun taas ei-toiminnalliset vaatimukset keskittyvät järjestelmän rajoituksiin, yleisiin piirteisiin ja ominaisuuksiin. Yhteistä vaatimuksille on kuitenkin se, että ne johdetaan käyttäjävaatimuksista.

Toiminnalliset vaatimukset määrittävät sen millaisia toimintoja järjestelmässä tulisi olla (Lamsweerden, 2009). IEEE standardin (1990) mukaan toiminnalliset vaatimukset kuvaavat suoritettavia toimintoja järjestelmälle tai jollekin järjestelmän komponentille. Toiminnallisia vaatimuksia voidaan kuvata myös järjestelmän pakollisiksi tehtäviksi (Pohl, 1994). Lamsweerden (2009) mukaan joissain tilanteissa on hankalaa erottaa toiminnallisten ja ei-toiminnallisten vaatimusten ero, sillä vaatimukset voivat olla osittain päällekkäisiä. Kun asiakkaan esittämää ei-toiminnallista vaatimusta muutetaan yksityiskohtaisemmaksi voi se muuttua helposti toiminnalliseksi vaatimukseksi, kun vaatimusta tarkennetaan tarpeeksi (Kotonya & Sommerville, 2002).

3.1.1 Vaatimusmäärittely ja vaatimusmäärittelyprosessi

Vaatimusmäärittely (*eng. requirement engineering, RE*) on yksi hyvin tutkittu aihealue ohjelmistokehitykseen liittyen. Vaatimusmäärittely koostuu useista eri vaiheista, joista jokainen jollain tavalla tähtää järjestelmän vaatimusten löytämiseen, analysointiin sekä dokumentointiin. (Haikala & Mikkonen, 2011, ss. 64-65; Lucia & Qusef, 2010, Kotonya & Sommerville, 2002) Vaatimusmäärittelyn avulla pyritään löytämään myös tekijöitä, joiden avulla ylläpidetään järjestelmää (Lucia & Qusef, 2010). Vaatimusmäärittelyn avulla pyritään tunnistamaan ohjelmistojen oikea käyttötarkoitus. Käyttötarkoitus voidaan tunnistaa analysoimalla ohjelmiston sidosryhmiä ja heidän tarpeitansa. Nämä tarpeet tulee dokumentoida, jotta kehitystyö, kommunikointi ja analysointi olisi onnistunutta. Vaatimusmäärittely keskittyykin keräämään, valmistelevaan ja muodostamaan määrittelydokumentin ennen suunnitteluvaihetta. (Lucia & Qusef, 2010; Kotonya & Sommerville, 2002)

Vaatimusmäärittely (RE) nähdään koostuvan systemaattisista ja toistuvista toimista, jotka varmistavat järjestelmävaatimusten kokonaisvaltaisuuden, johdonmukaisuuden ja tarkoituksenmukaisuuden. Muun muassa Lucia ja Qusef (2010), Paetsch et al. (2003), Kotonya ja Sommerville (2002), Darwish ja Megahed (2016) sekä Haikala ja Mikkonen (2011, s. 65) jakavat vaatimusmäärittelyn nelivaiheiseksi prosessiksi, jonka vaiheet on esitetty kuvassa 3.



Kuva 3. *Vaatusmäärittely prosessimalli (Lucia & Qusef 2010)*

Vaatusmäärittelyprosessin ensimmäinen vaihe on vaatimusten selvittäminen. Tässä vaiheessa kerätään vaatimuksia käyttäjiltä ja muilta sidosryhmiltä. Ensin määritetään mitä sidosryhmillä tarkoitetaan ja kenen tulee osallistua vaatusmäärittelyyn. (Haikala & Mikkonen, 2011, ss. 65-67; Kotonya & Sommerville, 2002) Kun sidosryhmät on määritetty, voidaan siirtyä selvittämään vaatimuksia (Quesf, 2010). Huomattavaa on, että kaikki vaatimukset eivät ole välttämättä edes sidosryhmien edustajille tiedossa ja näitä piilossa olevia vaatimuksia tulisi myös mallintaa mahdollisimman tarkasti. Tärkeintä tässä vaiheessa prosessia on tunnistaa järjestelmän rajoitteet, rajat sekä identifioida ongelmat. Yleisesti tätä vaihetta voidaan kutsua myös vaatimusten keräämiseksi. (Kotonya & Sommerville, 2002) Jotta ensimmäinen vaihe on onnistunut Quesf (2010) mukaan tässä vaiheessa tulisi vastata erityisesti kolmeen kysymykseen:

1. Edistääkö järjestelmä korkean tason tavoitteita?
2. Voidaanko järjestelmä toteuttaa annetussa budjetissa?
3. Onko järjestelmä toteutettavissa?

Prosessin toisessa vaiheessa alkuperäiset käyttäjien tarpeet ja vaatimukset tarkennetaan ja priorisoidaan. Analysointi auttaa tunnistamaan olennaiset tarpeet käyttäjien näkökulmasta sekä samalla tunnistaa konfliktit ja epä johdonmukaisuudet niiden välillä. (Eberlein & Maurer, 2003) Analyysivaiheessa Sutcliffe (2009) mukaan voidaan esittää viisi erilaista kysymystä, joihin vastaamalla vaatimusten tarkentaminen ja priorisointi on mahdollista. Näitä kysymyksiä ovat:

1. Mikä on järjestelmän tarkoitus (tavoitteet)?
2. Mitkä esineet ovat mukana?
3. Missä järjestelmä sijaitsee?
4. Milloin asioiden pitäisi tapahtua?
5. Miksi järjestelmä on välttämätön? (tavoitteet tai ongelmat, joita sen tulee ratkaista)

Dokumentointivaiheen tarkoituksena on muodostaa selkeä materiaali siitä, millaisia vaatimuksia sidosryhmien edustajilla on järjestelmälle (Haikala & Mikkonen, 2011, s. 66). Dokumentti toimii sidosryhmien ja kehittäjien välillä viestinnän työkaluna ja vaatimuksia voidaan esittää monella eri tavalla kuten käyttäjätarinoiden, käyttötapauksien, prototyyppien tai käsitteellisten kaavioiden ja toimialuemallin avulla (Ylikangas & Kauppinen, 2016; Hamlet & Maybee, 2002). Jos dokumentti on puutteellinen, voidaan pitkällä aikavälillä joutua ongelmiin, sillä järjestelemää kehittäessä ei välttämättä tiedetä, mitä sidosryhmät oikeasti haluavat järjestelmän käytöllä saavuttaa, jos sitä ei ole selkeästi ilmaistu dokumentissa (Quesf, 2010). Vaatimusdokumentin tulee olla selkeä, johdonmukainen, tiivis ja toteutettavissa oleva (Eberlein & Maurer, 2003).

Vaatimusmäärittelyn viimeisessä vaiheessa validoidaan vaatimuksia. Tarkoituksena on varmistaa, että vaatimukset ovat täydellisiä, johdonmukaisia ja selkeitä, jotka tyydyttävät kaikki sidosryhmät. (Elshabdidy & Mazen, 2013) Tässä vaiheessa tulee kommunikoida vaatimuksista asiaan kuuluvien sidosryhmien kanssa ja tarkistaa vaatimusten paikkaansa pitävyys ennen toteuttamista. Tämän tarkistusprosessin aikana havaintoja kerätään sidosryhmiltä ja niillä voidaan täydentää vaatimusmäärittelyn dokumenttia. Kun kerän validoitu vaatimusmäärittelyn prosessi on saatu päätökseen kaikki sidosryhmät ovat virallisesti hyväksyneet vaatimukset. (Robertson & Robertson, 2011; Haikala & Mikkonen, 2011, s. 67) Jos tilanteet kuitenkin muuttuvat tai esille nousee jotain ristiriitoja vaatimusten kesken, tulee niitä muokata. Tässä vaiheessa vaatimusmäärittely tulee uudelleen validoida uusilla vaatimuksilla. Vasta uuden validoinnin jälkeen muokattuja vaatimuksia voidaan käsitellä ja kehittää. (Paetsch et al. 2003; Elshabdidy & Mazen, 2013; Eberlein & Maurer, 2003)

Vaatimusten hallinta on kokonaisvaltainen prosessi aikataulujen, koordinoinnin ja dokumentoinnin hallintaan (Elshabdidy & Mazen, 2013). Sommervillen (2010) mukaan vaatimusten hallinta on prosessi järjestelmän vaatimusten muutosten ymmärtämiseksi ja hal-

litsemiseksi. Prosessi ottaa huomioon kaikki toiminnot, jotka liittyvät järjestelmän vaatimusten muuttumiseen. Erilaisia työkaluja ja tekniikoita käytetään prosessin yksinkertaistamiseen. Vaatimusten jäljitettävyyden on tärkeä osa vaatimusten hallintaprosessia (Nuseibeh & Easterbrook, 2000). Jäljitettävyyden määrittelee sen, kuinka yksinkertaista on lukea, navigoida sekä vaihtaa vaatimuksia vaatimusdokumentista. (Elshabdidy & Mazen, 2013) Yksi tärkeä tehtävä vaatimusten hallinnalla on muutosjohtaminen. Muutosjohtaminen viittaa kyvykkyyteen hallita muutoksia ja vaatimuksia läpi järjestelmäkehityksen elinkaaren. Jokainen muutos tai puuttuva tieto vaatimuksissa johtaa muutoksenhallinnan prosessiin. Tehokkaalla vaatimustenhallinnalla pystytään saavuttamaan liiketoiminnan tavoitteet tehokkaasti. (Pandey et al. 2010)

Kotonya ja Sommerville (2002) mukaan vaatimusmäärittelyprosessi sisältää erilaisia toimintoja, joiden avulla projekti etenee. Jokaisella kehitystiimin jäsenellä on oma rooli prosessissa. Erityisen tärkeänä Kotonya ja Sommerville (2002) pitävät sidosryhmien osallistumista vaatimusmäärittelyprosessiin ja sitä, kuinka erilaiset sidosryhmät vaikuttavat prosessin etenemiseen. Sidosryhmien edustajat voivat olla ohjelmistoalan ammattilaisia tai järjestelmän käyttäjiä, joilla jokaisella on jotain teknistä tietämystä järjestelmästä sekä omia henkilökohtaisia tavoitteita. Sidosryhmien edustajat voivat olla eri osastoilta organisaatiosta ja he ajavat omia tavoitteita ottamatta huomioon muiden tarpeita tai koko vaatimusmäärittelyprosessia. (Haikala & Mikkonen, 2011, ss. 66-68; Phol & Rupp, 2011) Tämän takia tulee saada jokaisen sidosryhmän edustajat osallistumaan vaatimusmäärittelyyn, jotta kaikki pystyvät vaikuttamaan prosessin etenemiseen omalla tavallaan ja varmistetaan siitä, että kaikki tarvittavat vaatimukset pystytään tunnistamaan.

Parantainen (2007, s. 210-225) jakaa vaatimusmäärittelyprosessin kolmeen vaiheeseen, joita ovat:

1. Listaa palvelun reunaehdot
2. Laadi palvelusta käsikirjoitus
3. Tarkenna vaatimukset käsikirjoituksen perusteella

Prosessin ensimmäisessä vaiheessa listataan palvelun reunaehdot eli kirjataan ylös kaikki tuotetta tai palvelua koskevat toiveet tai vaatimukset. Nämä voidaan usein johtaa asiakkaalle jo annetusta lupauksesta, mutta niihin voi vaikuttaa myös yrityksen strategiset linjaukset. (Parantainen, 2007, ss. 209-210) Tässä vaiheessa Parantaisen (2007, ss. 210-211) mukaan vaatimukset tulee järjestää karkeasti tärkeysjärjestykseen. Prosessin toisessa vaiheessa laaditaan tuotteesta tai palvelusta käsikirjoitus. Tässä vaiheessa

hahmotellaan palvelun kulku lyhyiden tarinoiden avulla. Asiakkaan näkökulma tulee pitää mielessä, vaikka tuotteiden tai palveluiden kuvaaminen tapahtuu lopulta yrityksen omassa formaatissa. Kun tuotteen tai palvelun kulku on hahmotettu tarinoiden avulla, siirrytään prosessin viimeiseen vaiheeseen, jossa tarkennetaan vaatimuksia käsikirjoituksen perusteella. Prosessin kolmas vaihe voi kestää useitakin kierroksia ja tässä vaiheessa joudutaan luultavasti palaamaan ensimmäiseen vaiheeseen useita kertoja. Tämä kuitenkin mahdollistaa sen, että laadittu vaatimusmäärittely toimii lähtökohtana seuraavassa vaiheessa laadittavalle työohjeelle. Työohjeissa kuvataan täsmällisesti vaatimusmäärittely kohta kohdalta sekä toteutettavan tuotteen tai palvelun vaiheet ja vastuut. (Parantainen, 2007, ss. 210-225)

Ketterälle projektille on tyypillistä se, että vaatimusmäärittelyä muutetaan projektin edetessä vastaamaan muuttuvia tilanteita. (Phol & Rupp, 2011) Tämän takia vaatimusmäärittelyä voidaan joutua muuttamaan useaan otteeseen sitä mukaan, kun tuotteen tai palvelun formaatti alkaa hahmottua. Ketterässä tuotteistamisessa on tarkoituksena palata tarkastelemaan ajoittain pitäisikö vaatimusmäärittelyä muuttaa johonkin suuntaan. (Phol & Rupp, 2011; Parantainen, 2007, ss.195-198) Parantaisen (2007, s. 197) mukaan liian kauaksi ennustaminen ei ole järkevää projektin onnistumisen kannalta ja jos kehitysprojektin aikana havaitaan uusia asiakastarpeita tai näkökulmia, tulee vaatimuksia muuttaa tilanteisiin sopivaksi.

Parantaisen (2007, ss. 196-198) mukaan hyvin laaditun vaatimusmäärittelyn avulla voidaan saavuttaa monia hyötyjä:

1. Varmistetaan, että lopputulos vastaa asiakkaan todellisia tarpeita.
2. Valmiin vaatimusmäärittelyn perusteella arvioidaan projektin laajuus kuten aikataulu ja kustannukset. Tuotteen ominaisuudet ovat johdettavissa vaatimusmäärittelystä ja niiden avulla tuotantokustannukset ovat ennustettavissa.
3. Useita rinnakkaisia osaprojekteja voidaan aloittaa yhtäaikaaisesti eikä tarvitse odottaa edellisten vaiheiden valmistumista.
4. Vaatimusmäärittelyssä listataan vaatimukset priorisointijärjestykseen. Kun vaatimukset ovat tärkeysjärjestyksessä pystytään löytämään vaadittu raja sille, mihin kehitysprojektin resurssit ja rahat riittävät. Vähemmän tärkeät vaatimukset voidaan jättää tuotteen seuraavaan versioon tai jatkokehitysvaiheeseen.
5. Säästää kuluja, sillä se karsii pois ominaisuuksia, joille ei löydy tarpeeksi perusteluja tai kysyntää.

6. Jos vaatimusmäärittely on toteutettu oikein tuottaa se palvelumittarit luonnostaan.
7. Testausvaiheen pohjana käytetään vaatimusmäärittelyä ja testien aikana tunnistetaan se, toimiiko järjestelmä vaaditulla tavalla.

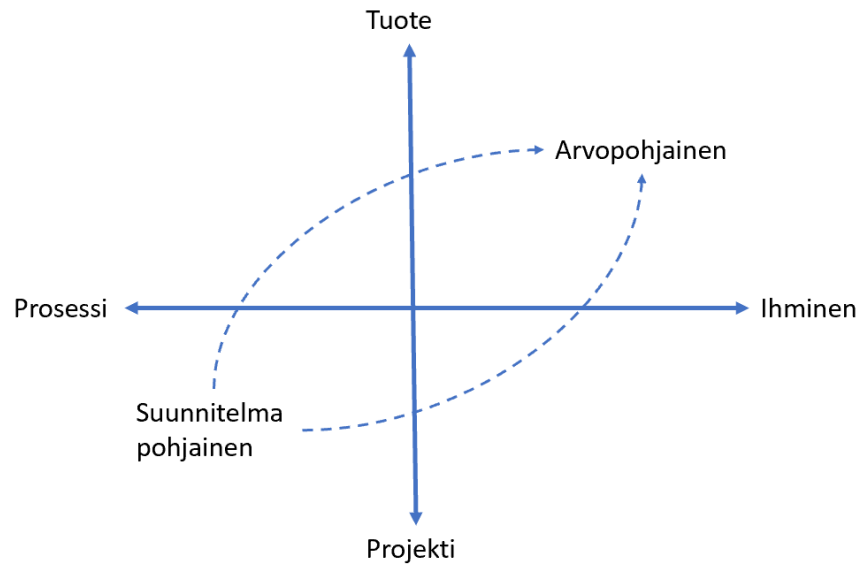
Huonona vaatimusmäärittelynä voidaan pitää sellaista, joka perustuu väärin joko keksittyihin tai oletettuihin ajatuksiin siitä, mitä asiakas tarvitsee (Parantainen, 2007, s. 198). Jos asiantuntija rajoittaa vaatimusmäärittelyä omilla oletuksillaan voidaan päätyä hyvin erilaiseen lopputulokseen mitä on haluttu. Jos vaatimukset jätetään perustelematta ei voida jälkikäteen tietää kuinka todellinen alkuperäinen vaatimus on ollut. (Haikala & Mikkonen, 2011, ss.65-66) Huomattavaa on myös Parantaisen (2007, s. 204) mukaan se, että hyvän vaatimusmäärittelyn tulee rajoittua vain vaatimuksiin. Vaatimusmäärittelyn aikana määritellä, mitä palvelun tulee saada aikaan eikä miten kehitettävän palvelun tulisi toimia. Jos vaatimusmäärittelyn aikana mietitään liikaa sitä, miten jokin järjestelmäominaisuus toteutetaan, saadaan lopputuloksena hyvin yksinkertaisia vaatimuksia ja jopa liian helppoja toteuttaa. Tällöin tulokset eivät ole kekseliäitä.

3.1.2 Ketterä vaatimusmäärittely

Kun mietitään ketterää ohjelmistoprojektia, projektin laajuus on määritelty etukäteen ja sitä tarkastellaan jokaisessa iteraatiossa. Vaatimukset on määritelty alun perin asiakkaiden kanssa ja listattu asiakkaan toivelistaksi. (Schön et al. 2016; Darwish & Megahed, 2016) Asiakas työskentelee läheisesti kehitystiimin kanssa, jotta vaatimusten määrittäminen on tehokasta ja jatkuvaa. Kehitysprosessi on dynaaminen ja se antaa tilaa uusien alueiden tunnistamiseen ja vaatimusten määrittämiseen koko kehitysprosessin ajan. (Schön et al. 2016)

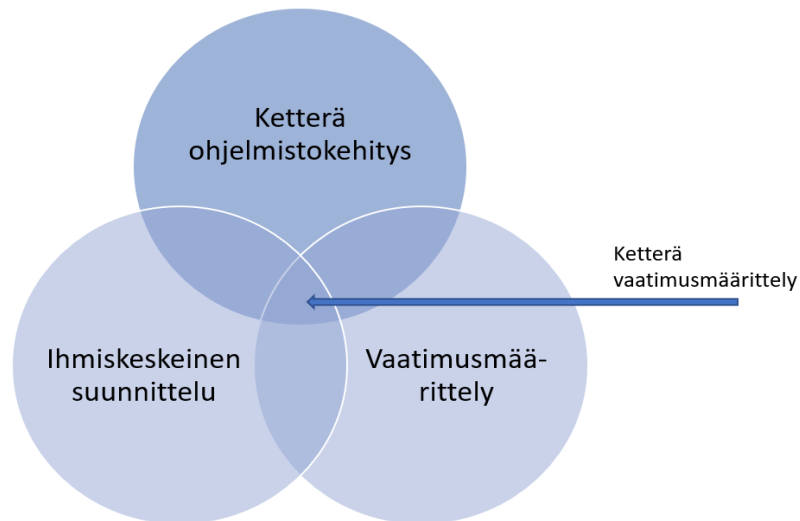
Ketterä ohjelmistokehitys on siirtynyt enemmän suunnitelmapohjaisesta kehittämisestä arvopohjaisempaan suuntaan. Tämä tarkoittaa sitä, että nykyisin suunnitteluprojekteissa painotetaan enemmän tuotetta ja ihmiskeskeisyyttä kuin ennen. Suunnitelmapohjainen kehittäminen perustui vahvasti prosesseihin ja niiden kautta pyrittiin mallintamaan ohjelmistoprojektin vaatimukset. (Schön et al. 2016) Kuvassa 4 on esitetty ketterän ohjelmistoprojektin muuttunut käsite. Suunnitelmapohjaiselle kehitykselle on tyypillistä, että sen aikana neuvotellaan hinnoittelumalleista, projektisuunnitelmasta ja siitä kuinka monta tuotetta tai ominaisuutta pystytään kehittämään saatavilla olevilla resursseilla. Arvopohjaisessa kehittämisessä ihmiset keskustelevat visioista, kokemuksista ja inhimillisistä ar-

vioista ja siitä kuinka näitä tekijöitä voidaan tuoda toteutettavaan tuotteeseen asti. Ketterässä ohjelmistokehityksessä prosessin tulisi lähteä aina liikkeelle ihmiskeskeisistä asioista ja inhimillisistä arvoista. (Schön et al. 2017)



Kuva 4. *Suunnitelmapohjainen kehittäminen muuttumassa arvopohjaiseksi (Schön et al. 2017)*

Ketterä vaatimusmäärittely on monialainen tutkimusalue, joka koostuu ketterästä ohjelmistokehityksestä, vaatimusmäärittelystä ja ihmiskeskeisestä suunnittelusta. Kuvassa 5 on nähtävissä ketterän vaatimusmäärittelyn monialaisuus ja suhde muihin tutkimusalueisiin. (Schön et al. 2017) Cao & Ramesh (2008), Rames et al. (2010), Boness ja Harrison (2007) sekä Wolfgang (2011) esittävät, että perinteisen vaatimusmäärittelyn (*eng. requirements engineering*) vaiheita tulisi käyttää myös ketterien menetelmien kanssa, sillä yhdistämällä ne ihmiskeskeisen suunnittelun periaatteisiin ja ketterään ohjelmistokehitykseen saavutetaan ketterä vaatimusmäärittely. Ketterään vaatimusmäärittelyyn voidaankin sisällyttää vaatimusmäärittelyn prosessimallin vaiheita kuten vaatimusten selvittämistä, analysointia, dokumentointia tai validointia. Zhu (2009) mukaan olisi myös tärkeää hyödyntää vaatimusmäärittelyn käytäntöjä kuten tarkkailua, haastatteluita, työpajoja sekä vahvaa yhtenäistä tekemistä ketterässä vaatimusmäärittelyssä.



Kuva 5. *Ketterä vaatimusmäärittelyn monialaisuus*

Vaikka kirjallisuudessa on jo laajasti tunnistettu ketterien tapojen mukaisia menetelmiä, jotka sopisivat vaatimusmäärittelyn prosessimalliin ei vielä tiedetä tarpeeksi vaatimusmäärittelyn prosesseista tai käytännöistä, jotka sopivat hyvin joustaviin ja dynaamisiin työtapoihin. (Inayat et al. 2015)

Lucia ja Qusef (2010) esittävät artikkelissaan sen, miten vaatimusmäärittelyprosessin vaiheet voidaan implementoida scrum-projektimenetelmään. Nämä implementointi menetelmät on kuvattu taulukossa 2. Scrum -projektimenetelmä tarjoaa viitekehyksen ketterälle ohjelmistoprojektille ja sen toteutukselle. Scum keskittyy maksimissaan 30-päivän sprinttien sykleihin, joiden toteutettavat vaatimukset ovat lähtöisin tuotteen kehitysjonolta. (Schwaber & Sutherland, 2017)

Taulukko 2. *Vaatusmäärittelyn implementointi ketteriin projektimenetelmiin (Lucia & Qusef 2010)*

<i>Vaatusmäärittelyn aktiviteetti</i>	<i>Scrum implementointi</i>
<i>Vaatususten selvittäminen</i>	<ul style="list-style-type: none"> • Tuoteomistaja muodostaa tuotteen kehitysjonon • Kaikki sidosryhmät ja niiden edustajat osallistuvat tuotteen kehitysjonon muodostamiseen • Vaatusukset kuvataan tarinoina

<i>Analysointi</i>	<ul style="list-style-type: none"> • Kehitysjonon tarkennuskokous • Tuoteomistaja priorisoi tuotteen kehitysjonolla olevat vaatimukset • Tuoteomistaja analysoi vaatimusten toteutettavuuden
<i>Dokumentointi</i>	<ul style="list-style-type: none"> • Kasvotusten tapahtuvaa kommunikointia
<i>Validointi</i>	<ul style="list-style-type: none"> • Tarkastelutapaaminen (eng. review meeting) • Tiheää palautteen vaihtoa
<i>Vaatimusten hallinta</i>	<ul style="list-style-type: none"> • Sprintin suunnittelutapaaminen • Tuotteen kehitysjonolla olevat nimikkeet (käyttäjätarinat) tarkastelun kohteeksi • Muutokset vaatimuksissa lisätään tai poistetaan tuotteen kehitysjonolta

Suurin ero tavallisen vaatimusmäärittelyn (eng. requirements engineering) ja ketterän vaatimusmäärittelyn (eng. requirements engineering) välillä on se, että perinteinen vaatimusmäärittely pyrkii mallintamaan vaatimukset dokumentin muodossa ennen suunnitteluvaihetta. Ketterä vaatimusmäärittely puolestaan huomioi muuttuneita vaatimuksia myös paljon myöhemmissä vaiheissa kehitysprosessia sekä kehittää toimintaa niiden perusteella. (Ramesh et al 2010; Schön, 2017; Lucia & Qusef, 2010)

Lucia ja Qusef (2010) esittävät työssään kymmenen erilaista ohjetta siihen, miten ketterää vaatimusmäärittelyä voidaan parantaa. Nämä ohjeet on koottu taulukkoon 3. Onnistuneessa ketterässä vaatimusmäärittelyssä vuorovaikutus asiakkaiden kanssa tulee olla sujuvaa ja säännöllistä. Ei-toiminnalliset vaatimukset tulee myös tunnistaa ennen kuin ketterää ohjelmistoprojektia voidaan aloittaa, sillä niiden merkitys tulee nähdä yhtä tärkeä kuin toiminnallisten vaatimusten. Tässä työssä keskitytään toiminnallisten vaatimusten määrittämiseen ja siksi ei-toiminnalliset vaatimukset on rajattu pois. Lucia ja Qusef (2010) nostavat myös esille sen kuinka tärkeää on miettiä, minkä kokoinen projektitiimin tulee olla.

Taulukko 3. Ohjeita ketterään vaatimusmäärittelyyn (Lucia & Qusef 2010)

Ohje	Määritelmä
<i>Asiakkaan osallistaminen</i>	Koska ketterä ohjelmistokehitys keskittyy vahvasti vuorovaikutukseen asiakkaan kanssa. Ennen projektin aloitusta kaikki tärkeät ja relevantit sidosryhmät tulee tunnistaa ja heidän vaatimuksiaan tulee kunnioittaa. Erilaisen työkalujen ja menetelmien avulla pyritään keräämään niin paljon informaatiota sidosryhmiltä kuin mahdollista, jotta voidaan ratkaista kaikki epä johdonmukaisuudet.
<i>Ketterät projektisopimukset</i>	Projektin alussa kaikista kriittisimmät vaatimukset tulee ilmaista sidosryhmien avulla niin hyvin kuin mahdollista. Tämän avulla kokenut projektijohtaja pystyy määrittämään ketterän hankkeen kustannukset sekä arvioimaan millaisia kustannuksia syntyy muutoksista projektin myöhemmissä vaiheissa.
<i>Tiheät julkaisut</i>	Säännöllisesti ja tiheään tapahtuvat julkaisut antavat mahdollisuuden julkaista nopeammin haluttuja muutoksia asiakkaalle ja heiltä on mahdollista saada palautetta toteutetuista ominaisuuksista.
<i>Vaatimusten määrittelyn kieli</i>	Käytä kielitieteellisiä menetelmiä vaatimusten aikaansaamiseksi. Vaatimukset kerätään käyttämällä asiakkaan kieltä eikä formaalia kieltä.
<i>Ei-toiminnalliset vaatimukset</i>	Ei-toiminnallisista vaatimuksista tulee keskustella erillisissä kokouksissa varhaisessa vaiheessa. Kun ei-toiminnalliset vaatimukset on tunnistettu ja dokumentoitu, voi kehitystiimi aloittaa järjestelmän kehittämisen.
<i>Pienemmät tiimit ovat joustavampia</i>	Pienemmät tiimit mahdollistavat jatkuvan kommunikoinnin tiimin ja sidosryhmien välillä tehokkaasti. Samalla sopivan kokoinen tiimi mahdollistaa tehokkaan muutosjohtamisen.
<i>Vaatimusten evoluutio</i>	Ketterä vaatimusmäärittely mukautuu vaatimuksiin myös myöhemmissä vaiheissa kehitysprojektia. Muuttuvien vaatimusten kohdalla täytyy kuitenkin toteuttaa joka kerta oma iteraatiokierros, jotta saadaan hyväksyntä sidosryhmiltä. Tämä voi pitkittää projektia hetkellisesti ja siksi tuleekin varmistaa, että iteraatiot ovat yhdenmukaisia sekä odotukset ja kehitysprosessi pysyvät järjestelmällisenä.
<i>Ei aikaista dokumentointia</i>	Aikaisessa vaiheessa toteutetut dokumentit voivat muuttua hetkessä epä-relevanteiksi sillä ketterä kehitys kannustaa vaatimusten muuttumiseen ja siihen reagointiin. Tämän takia dokumentointia ei kannata suorittaa liian aikaisessa vaiheessa kehitysprosessia.
<i>Vaatimusten jakaminen osiin</i>	Jos kehitystiimi kokee jonkin vaatimuksen liian kompleksiseksi, tulee se jakaa pienempiin helpommin käsiteltäviin osiin. Tämä helpottaa tiimiä ymmärtämään paremmin toiminnallisia vaatimuksia, joita asiakkaalla on ja

	mahdollistaa sujuvamman kommunikoinnin tiimin sisällä. Esimerkiksi käyttäjätarinoin mallinnetut vaatimukset jaetaan lyhyempiin tarinoihin.
<i>Vaatimusten jäljitettävyyys</i>	Kun vaatimukset muuttuvat kehitysprojektin aikana on tärkeää varmistaa, että järjestelmän suunnittelu vastaa nykyistä vaatimusten joukkoa. Vaatimusten tulisi olla jäljitettävissä, jotta tiimi pystyisi uudelleen rakentamaan jonkin osan vaatimusmäärittelyn perusteella.

Schön et al. (2017) tunnistivat tutkimustaan tehdessään erilaisia ketteriä menetelmiä, joita käytetään sidosryhmien osallistamiseksi, datan keräämisen tukena ja ihmiskeskeisen suunnitteluprosessin sekä ketterän ohjelmistokehityksen integroimisessa. Tutkijat löysivät 20 erilaista artefaktia ja menetelmää, joita hyödynnettiin kirjallisuuskatsauksesta tehdyissä artikkeleissa. Näitä artefakteja ovat muun muassa: käyttäjätarinat, prototyypit, käyttötapaukset, skenaariot, tarinakortit, persoonat, visiot, kuvakäsikirjoitukset, tehtävät, kanban taulut, ajatuskartat ja videot. Schön et al. (2017) nostivat tutkimuksessaan esille myös sen, että ihmiskeskeinen suunnittelu ja palvelumuotoilu ovat usein käytettyjä metodeja kehittämään ketterää ohjelmistokehitystä enemmän ihmiskeskeisempään suuntaan.

3.1.3 Vaatimusten esillesaanti, kuvaaminen ja priorisointi

Ennen kun vaatimuksia voidaan kuvata tulisi ne saada jollain tavalla esille. Esillesaannin tueksi on kehitetty erilaisia tekniikoita, joita hyödyntämällä voidaan tunnistaa sidosryhmien tarpeita. Tekniikat vähentävät vaatimusten monimutkaisuutta ja lisäävät selkeyttä. (Chua et al. 2010) Tekniikoita voidaan luokitella erilaisiin ryhmiin esimerkiksi projektivaiheiden kautta. Nuseibehin ja Eastbrookin (2000) luokittelevat vaatimusten esillesaannin kuuteen eri ryhmään, joita ovat perinteinen tekniikka, prototyypit, ryhmätekniikat, kontekstuaaliset tekniikat, kognitiiviset tekniikat ja mallilähtöiset tekniikat. Taulukossa 4 on nähtävissä luokitellut tekniikat vaatimusten esillesaantiin.

Taulukko 4. *Vaatimusten esillesaannin tekniikoita (Nuseibehin & Eastbrook 2000)*

Luokittelu	Erilaisia tekniikoita
<i>Perinteinen</i>	Kysely, haastattelu ja dokumenttianalyysi
<i>Prototyyppi</i>	kertakäyttöinen ja kehittyvä prototyyppi
<i>Ryhmätyöskentely</i>	Aivoriihi ja kohderyhmä
<i>Kontekstuaalinen</i>	Etnografia, keskusteluanalyysi ja kontekstuaalinen kysely
<i>Kognitiivinen</i>	Protokolla-analyysi, porrastaminen ja korttien-lajittelu

Schön et al. (2016) toteuttivat tutkimuksen, jossa tarkasteltiin erilaisia artefakteja ketterässä vaatimusmäärittelyssä. Heidän tutkimuksensa perusteella esille nousi viisi erilaista tekniikkaa, joiden merkitys ketterän vaatimusmäärittelyn toteutuksessa on erityisen tärkeä. Näitä menetelmiä ovat käyttäjätarinat, prototyypit, käyttötarinaskenaariot, story card ja persoonat. Tekniikoiden määritelmät on esitetty taulukossa 5.

Taulukko 5. Tekniikat ketterään vaatimusmäärittelyyn (Schön et al. 2016)

Artefakti	Määritelmä
<i>Käyttäjätarinat</i>	Piirteet kuvataan henkilön näkökulmasta, joka tarvitsee vaaditun ominaisuuden.
<i>Prototyypit</i>	Malli ohjelmistosovelluksesta, joka tukee suunnitteluvaihtoehtojen arviointia ja kommunikointia.
<i>Käyttötarinaskenaariot</i>	Käyttötarinat kuvaavat toimintoja tai vaiheita, joita tarvitaan, jotta saavutetaan tavoitteet. Skenaario on tekstimuotoinen esitys ongelmasta ja se kuvailee vuorovaikutusta käyttäjän ja järjestelmän välillä tietyssä kontekstissa.
<i>Tarinakortti</i> (eng. <i>story card</i>)	Fyysinen esitys kirjoitetusta tekstistä ja yksityiskohdista käyttäjätarinoissa.
<i>Persoonat</i>	Kuvaelma kuvitteellisesta persoonasta, joka edustaa suurempaa kohde käyttäjäryhmää.

Kun vaatimukset on saatu esille, voidaan niitä kuvata eri keinoin. Yleistä on, että ketterässä vaatimusmäärittelyssä vaatimukset kuvataan käyttäjätarinoiden muodossa. Scrum projektimenetelmälle on tyypillistä, että tuoteomistaja on vastuussa käyttäjätarinoiden kirjoittamisesta, kun taas XP menetelmässä asiakkaat kirjoittavat käyttäjätarinat. (Leffingwell, 2011, ss. 58-61) Käyttäjätarinat voivat alkuun olla hyvin laajoja kuvauksia vaatimuksista, mutta toteutusvaiheessa tarinoita tarkennetaan yksityiskohtaisiksi tehtäviksi (Leffingwell, 2011, ss. 58-61; Zhang et al. 2010).

Käyttäjätarinat ovat laajasti käytettyjä mallinnuksia vaatimuksista ketterässä kehityksessä. Ne ovat lähtöisin ketterästä XP menetelmästä, josta se on otettu käyttöön myös muissa projektimenetelmissä. Aluksi käyttäjätarinoiden avulla kuvattiin erilaisia käyttötapauspaus. Nykyisin kuitenkin käyttäjätarinoiden ja käyttötapausten välinen ero on tullut

selvemmäksi ja kyseessä on kaksi erilaista tekniikkaa vaatimusten mallintamiseen. Käyttäjätarinat ovat pieniä palasia liiketoiminnan arvosta, jota voidaan toimittaa projektin iteraatioiden myötä. (Girvan & Paul, 2017, ss. 182-184) Käyttäjätarinoiden tarkoituksena on kuvata toiminnalliset vaatimukset niin, että kuvaukset ovat mahdollisimman lyhyitä, mutta hyvin selkeitä. Toiminnallisilla vaatimuksilla tarkoitetaan asioita, joita järjestelmän kuuluu toteuttaa siinä vaiheessa, kun järjestelmä on valmis. (Phol & Rupp, 2011; Lucasen et al. 2015) Tarinat voidaan nähdä työkaluina käyttäjien ja kehittäjien välillä sillä ne määrittelevät järjestelmän käyttäytymistä ymmärrettävässä muodossa. Käyttäjätarinoissa tulee mallintaa käyttäjän saamaa hyötyä ja arvoa. Yksinkertaisimmillaan käyttäjätarinat muodostavat listan vaatimuksista, joita järjestelmän tulee suorittaa. (Lefingwell, 2011)

Girvan ja Paul (2017) määrittelevät, että käyttäjätarinoita voidaan jakaa kolmeen eri kategoriaan niiden koon ja tason mukaan. Näitä kategorioita ovat teema, epiikat ja käyttäjätarina. Taulukossa 6 on esitetty tasot käyttäjätarinoiden luokittelulle.

Taulukko 6. Käyttäjätarinoiden kategorisointi (Girvan & Paul 2017)

Kategoria	Määritelmä
<i>Teema</i>	Teemat ovat loogisia ylätasoon käsitteitä, joita annetaan erilaisille käyttäjätarinoiden ryhmille. Käyttäjätarinat tallennetaan ja jaotellaan ryhmiin, jotta niiden organisointi ja käsittely olisi helpompaa. Ryhmien tulee olla loogisia ja ne voivat perustua käyttäjärooleihin tai ominaisuuksien ympärille. Teemat pyrkivät olemaan ylätasoon kuvauksia, jotka kuvaavat tärkeimpiä tavoitteita. Teemoja ei tule kirjoittaa käyttäjätarinoiksi, vaan ne ovat vain suuntia, joiden alla käyttäjätarinoita säilytetään.
<i>Epiikat</i>	Epiikat on nimi, joka annetaan laajemmille käyttäjätarinoille ja ne ovat yleensä helposti identifioitavissa ja vaikeasti arvioitavissa. Epiikat ovat isoja kokonaisuuksia ja niille ei ole välttämättä määritetty haluttua lopputulosta tai -pistettä. Epiikat tulee jaotella pienempiin kokonaisuuksiin, jonka jälkeen niitä ei enää tarvita kehitysvaiheessa. Epiikoita kutsutaan usein merkittäviksi aloitteiksi, jotka auttavat ohjaamaan arvoketjua kohti isomman portfolion tavoitteita. Ne ajavat taloudellista arvoa yritykselle.
<i>Käyttäjätarina</i>	Käyttäjätarinat ovat lyhyitä kuvauksia halutuista toiminnallisuuksista, jotka kerrotaan käyttäjän tai asiakkaan näkökulmasta. Kun toiminnallisuus toimitetaan, tulisi sen tuottaa arvoa käyttäjälle tai asiakkaalle. Käyttäjätarinoiden tulisi olla niin pieniä, että ne pystytään toimittamaan yhden iteraation aikana. Käyttäjätarinat ovat ainoa kategoria, jotka toteutuvat toimivassa ohjelmistossa.

Käyttäjätarinoita laatiessa on tärkeää miettiä missä muodossa ja kuinka laajasti tarinoita tulee esittää. Liian yksityiskohtainen esitys hankaloittaa järjestelmän kehitystä ja liian tiiviisti ilmaistut tarinat voivat jättää jotain hyvinkin olennaista pois kehitysvaiheessa. (Girvan & Paul, 2017, ss. 182-185) Leffingwellin (2011) mukaan käyttäjätarinat tulisi mallintaa niin tarkasti, että tarinat voidaan toteuttaa muutaman viikon aikana ja niiden työmääräarviot tulisi olla helposti arvioitavissa. Usein kuitenkin tarinat ovat huonosti kirjoitettuja ja niihin liitetään paljon laadullisia ongelmia. Käyttäjätarinoiden tulisi noudattaa ennalta määritettyä muotoa. Käyttäjätarinoiden avulla pyritään mallintamaan mahdollisimman selkeästi sidosryhmien ja käyttäjien tarpeita tavoitteiden avulla. (Lucassen et al. 2015; Dalpiaz & Brinkkemper, 2018)

Kun käyttäjätarinoita mallinnetaan, kohdataan usein tilanne, jossa käyttäjätarinat ovat liian suuria toteutettavaksi. Suurien tarinoiden kestoja sekä vaadittavia resursseja on vaikea arvioida ja siksi niitä tulee jakaa pienempiin helpommin käsiteltäviin osiin. (Cohn, 2005, s. 121; Leffingwell, 2011, s. 111).

Waken (2001) sekä Girvan ja Paulin (2017) mukaan käyttäjätarinoita laatiessa tulisi miettiä kuutta eri ominaisuutta, jotta tarinoista saadaan muodostettua toimivia kokonaisuuksia. Näitä ominaisuuksia ovat itsenäisyys, neuvoteltavissa oleva, hyödyllisyys, estimoitava, pieni ja testattava. Taulukossa 7 on kuvattu ominaisuudet ja niihin liittyvät tärkeimmät piirteet ajatellen hyviä käyttäjätarinoita.

Taulukko 7. Käyttäjätarinoiden laadulliset ominaisuudet (Girvan & Paulin, 2017; Waken, 2001)

Ominaisuus	Määritelmä
<i>Itsenäinen</i>	Jokaisen tarinan tulisi olla itsenäinen niin, että se ei ole riippuvainen muista käyttäjätarinoista. Jos tarina ei ole itsenäinen voi se aiheuttaa ongelmia myöhemmissä vaiheissa kuten suunnittelussa ja arvioinnissa. Jokaisen käyttäjätarinan tulisi sisältää jokin tavoite ja luoda arvoa käyttäjälle.
<i>Neuvoteltavissa oleva</i>	Käyttäjätarinat eivät saa olla sopimuksia tai vaatimuksia, joita ohjelmistoon tulee implementoida. Tarinat ovat lyhyitä kuvauksia toiminnallisuuksista, joita tutkitaan ja joista neuvotellaan asiakkaan ja kehitystieteen välillä.

<i>Hyödyllinen</i>	Käyttäjätarinan tulee olla hyödyllinen käyttäjälle tai asiakkaalle. Sen tulee edustaa tavoitteen tai tuloksen arvoa, jonka asiakas tai loppukäyttäjä voi ymmärtää. Käyttäjätarinat, jotka ovat hyödyllisiä vain kehitystiimille eivät ole oikeanlaisia tarinoita. Käyttäjätarinoita kirjoitettaessa tulee ymmärtää seuraukset mutta ei sisällyttää niitä tarinaan.
<i>Estimoitava</i>	Kaikille tarinoille tulee antaa arvio. Jos niitä ei voida arvioida ne ovat joko liian isoja tai tiimiltä puuttuu jokin oleellinen tekniikka tai tieto arvioida tai tehdä laskelmia tarinaan liittyen.
<i>Pieni</i>	Jokaisen käyttäjätarinan tulee olla niin pieni, että ne ovat arvioitavissa työmäärän ja kustannusten suhteen. Projektin varhaisessa vaiheessa tarinat voivat olla hyvin laajoja sekä edustaa korkeantason tavoitetta ratkaisulta. Projektin edetessä tarinoita tulee kuitenkin harjoittaa pienempiin osiin, jotta ne voidaan toimittaa iteraation aikana.
<i>Testattava</i>	Jokaista käyttäjätarinaa tulee pystyä testaamaan. Tämä vaihe vahvistaa sen, että ohjelmisto on toteutettu oikein. Ketterissä projekteissa testaamista tulisi automatisoida niin paljon kuin mahdollista.

Käyttäjätarinoita kirjoitetaan usein lapuille ja niissä kuvataan lyhyesti jokin haluttu toiminto, joka järjestelmän tulee täyttää. Yksinkertaisimmillaan näistä käyttäjätarinoista muodostuu lista toiminnoista tai ominaisuuksista, joita järjestelmältä vaaditaan. Käyttäjätarinoiden tulisi kuvata loppukäyttäjän kannalta arvoa tuottavia toiminnallisuuksia. Käyttäjätarinoita ei tarvitse säilyttää toteutuksen jälkeen ja siksi ne tarvitsevatkin vain vähän ylläpitoa. (Leffingwell, 2011, ss. 101-102) Leffingwell (2011, ss. 101-102) nostaa esille, että käyttäjätarinat eivät ole yksityiskohtaisesti määritettyjä vaatimuksia vaan niiden tulee olla neuvoteltavissa olevia ilmaisuja. Hänen mukaansa käyttäjätarinoilla on myös yleinen muoto, joka koostuu roolista, toiminnoista sekä arvosta, jota vaatimus tuottaa liiketoiminnalle. Myös Collier (2011, ss. 88-89) nostaa esille, että käyttäjätarinat tulee määrittää yleisen muodon mukaan, joka on:

Ollessani <rooli> voin <toiminto>, jotta <arvo liiketoiminnalle>.

Roolilla tarkoitetaan sitä henkilöä, joka tekee toiminnon tai sitä kuka saa arvoa toiminnon suorittamisesta. Toiminto tarkoittaa järjestelmän suorittamia toimintoja ja liiketoiminnan arvo puolestaan tarkoittaa sitä, mitä arvoa voidaan toiminnon kautta saavuttaa. (Leffingwell, 2011, s. 102) Leffingwell (2011, s. 102) mukaan käyttäjätarinarakenteen käytöstä hyvänä esimerkkinä voidaan pitää alla olevaa muotoa:

Kuluttajana <rooli> haluan nähdä päivittäisen sähkönkulutuksen <toiminto (mitä teen järjestelmällä)>, jotta voin vähentää sähkönkulutusta <arvo, jonka saavutan>.

Käyttäjätarinoita laatiessa on muistettava, että käyttäjätarinoilla ei tule ottaa kantaa käytettäviin teknologioihin tai dataan, jota ratkaisu tarvitsee. Collierin (2011, s. 88) ja Hugesin (2008, s.117) mukaan käyttäjätarinoiden tulee olla toteamuksia toiminnasta. Näiden toteamuksien tulee edustaa käyttäjän näkökulmaa ja ne tulee sitoa johonkin tiettyyn ongelmaan tai tarpeeseen. Käyttäjätarinat muodostuvat helposti liian monimutkaisiksi sillä asiakkaiden tarpeita ei ole helppoa mallintaa yksinkertaisella tavalla. Collierin (2001, ss. 89-91) mukaan projektitiimi on vastuussa siitä, että laajat käyttäjätarinat jaetaan pie-nempiin helpommin käsiteltäviin kokonaisuuksiin. Tämä vaatii tiimin jäseniltä taitoa käsitellä isompia kokonaisuuksia sekä tietoa, millainen toiminto pystytään toteuttamaan yhden sprintin aikana. Samalla hän huomauttaa, että käyttäjätarinat ovat tarkoitettu edustamaan vaatimuksia eikä niiden tule olla itse vaatimuksia.

Ambler (2002), Cohn (2004) ja Collier (2011) korostavat, että ketterässä kehityksessä olisi tärkeää mallintaa käyttäjätarinoita vasta juuri ennen sprintin alkua eli JIT (just-in-time) -periaatteen mukaisesti. Mitä ajankohtaisempia tarpeita pystytään mallintamaan käyttäjätarinoissa sen parempi.

Kun laaditaan vaatimusmäärittelyä toteutettavalle ohjelmistoprojektille, tulee erityisesti kiinnittää huomiota epämääräisiin sanoihin ja ilmaisuihin. Tällaisia epämääräisiä sanoja tai ilmaisuja ovat esimerkiksi joustava, turvallinen, mukautuva, kevyt, edullinen, helpokäyttöinen, riittävä, tarvittaessa, ja, tai, jne., yms. sekä normaalisti. Jokainen näistä ilmaisuista on tulkittavissa monella eri tavalla. Se mikä on toiselle helpokäyttöinen voi toiselle olla mahdoton käyttää. Esimerkiksi ilmaisu ”riittävän nopeasti” voidaan tulkita riippuen ihmisestä täysin eri tavalla. (Parantainen, 2007, ss. 206-208; Garret, 2011) Parantaisen (2007, s. 207) mukaan tämän selittää se, että kirjoittaja ei ole jaksanut tai halunnut määrittää sen tarkemmin mitä sana ”riittävä” tarkoittaa. Tässä voi olla kyse myös uskalluksesta ja ei haluta ottaa riskiä, sillä tilanteet ja asiakkaat ovat kaikki erilaisia. Parantainen (2007, ss. 207-208) nostaa esille, että vaatimusmäärittelyyn tulisi kirjoittaa juuri tällaisia päätöksiä, vaikka ne usein ovatkin harkintaan pohjautuvia arvauksia. Arvausten hyödyt voivat olla paljon suuremmat kuin avoimeksi jätetyt päätökset.

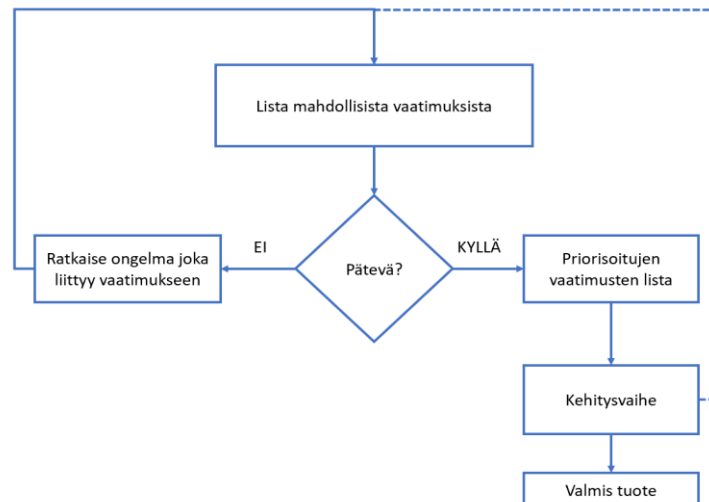
Onnistuneessa vaatimusmäärittelyssä tulee välttää passiivimuotoa sillä se antaa helposti kuvan, että määrittelijä ei suostu tekemään päätöksiä siitä kuka tekee mitään. (Gar-

ret, 2011) Vaatimusmäärittelystä on myös helppo unohtaa hyvin oleellisiakin vaatimuksia. Toisaalta määrittelystä voidaan saada liian laaja uppoutumalla yksityiskohtiin. Parantainen (2007, s. 207) korostaakin harjoituksen ja kokemuksen merkitystä onnistuneessa vaatimusmäärittelyssä. Hänen mukaansa myös vaatimusmäärittelyssä harjoituksen avulla voidaan saavuttaa parempia tuloksia.

Vaatimusten priorisointi

Vaatimusten priorisoinnilla tarkoitetaan sitä, että kaikista tärkeimmät ja oleelliset vaatimukset valitaan sekä päätetään missä järjestyksessä näitä vaatimuksia toteutetaan. Koska projektien budjetit ja aikataulut ovat aina rajalliset tulee vaatimusten kohdalla tarkasti miettiä mitä asioita koetaan hyödyllisiksi ja mitkä toiminnot toteutetaan ensin. (Paetsch et al. 2003) Kehitystiimin ja asiakkaan välillä on hyvä ylläpitää jatkuvaa keskustelua siitä, mitkä vaatimukset luovat asiakastyytyväisyyttä ja arvoa. Kehitystiimin vastuulla on myös kertoa asiakkaalle mahdollisista tekijöistä, jotka vaikuttavat vaatimusten priorisointiin. Tällaisia tekijöitä voivat olla esimerkiksi erilaiset riskit ja kustannukset. (Paetsch et al. 2003; Leffingwell, 2011) Vaatimusmäärittelyssä ensimmäisessä vaiheessa priorisoidaan epiikat ja ominaisuudet. Scrumin edetessä priorisoinnin kohteena on tuotteen kehitysjonolla olevat käyttäjätarinat (*eng. product backlog*). Priorisointi jatkuu myös siinä vaiheessa, kun tuotteen kehitysjonolta päätetään ottaa joitain käyttäjätarinoita seuraavan iteraation työlistalle eli sprintin kehitysjonolle (*eng. sprint backlog*). (Paetsch et al. 2003) Leffingwell (2011) muistuttaa, että vaatimusten priorisointia tulee jatkaa jokaisen iteraation jälkeen, kun uusia vaatimuksia selviää.

Lucian ja Quesfin (2010) mukaan vaatimusten priorisoinnille on tunnistettavissa oma prosessimalli. Tuoteomistajat ovat vastuussa vaatimusten priorisoinnista ja se vaatii jatkuvaa kommunikointia tiimin ja sidosryhmien kanssa. Kommunikoinnin avulla saadaan esille vaatimuksia, jotka täytyy toteuttaa ja vaatimuksia, joita on kiva omistaa. (Darwish & Megahed, 2016) Kuvassa 6 on esitetty vaatimusten priorisoinnin viitekehys.



Kuva 6. Vaatimusten priorisoinnin viitekehys (Lucia & Quesf 2010)

Priorisointi lähtee liikkeelle mahdollisten vaatimusten listasta, jolle aletaan tekemään yksittäisiä iteraatiokierroksia. Vaatimuksia tarkastellaan yksi kerrallaan ja arvioidaan ovatko vaatimukset päteviä vai liittyykö niihin jotain erityisiä ongelmia kuten päällekkäisyyttä tai liian suurta laajuutta. Jos vaatimus hyväksytään, voidaan se listata priorisoidujen vaatimusten listalle. Jos vaatimusta ei hyväksytä tulee sitä muokata, jotta vaatimus täyttää priorisoinnin kriteerit. (Lucia & Qusef, 2010)

Priorisoinnille on tunnistettavissa erilaisia periaatteita. Rachevan et al. (2010) mukaan vaatimuksia tulisi priorisoida sen mukaan kuinka paljon vähemmän arvoa ohjelmisto tuottaisi, jos kyseistä vaatimusta ei toteuteta järjestelmälle. Harris ja Cohn (2006) puolestaan perustavat priorisoinnin siihen, kuinka paljon ominaisuudet aiheuttavat muutostokustannuksia. Heidän mukaansa jokaiselle ominaisuudelle tulee arvioida niistä aiheutuvat muutostokustannukset ja sen perusteella niitä voidaan priorisoida. Leffingwell (2011, s. 209) puolestaan esittää, että vaatimusten priorisointiin yleisellä tasolla vaikuttavat monet tekijät ja siksi ominaisuuksien priorisointi perustuu taloudellisiin hyötyihin.

Priorisoitua vaatimuslistaa käytetään scrum-projektimenetelmän lähtökohtana. Tuotteen kehitysjono (*eng. product backlog*) on priorisoitu lista ominaisuuksista, toiminnoista, parannuksista, teknologioista ja vikakorjauksista, jotka kuvastavat tehtävää työtä. Tuotteen kehitysjono sisältää kaikki vaatimukset tuotteelle. (Schwaber & Sutherland, 2017) Lista on dynaaminen ja muuttuu riippuen siitä, millaista liiketoimintaa asiakas harjoittaa ja millaisia vaatimuksia heillä on. Ketterälle ohjelmistoprojektille on tyypillistä, että vaatimukset

muuttuvat projektin edetessä. Tämän takia tuotevastaavan tehtävänä on päivittää priorisoitujen vaatimusten listaa sekä pitää huolta, että vaatimukset toteutuvat niille määritettyjen sprinttien aikana. (Kotaiah & Khalil, 2017; Abdulhalim et al. 2018; Schwaber, 1997)

3.2 Käyttäjäkokemus ja ihmislähtöinen suunnittelu

Käyttäjäkokemus (*eng. user experience, UX*) on hyvin monimutkainen käsite, joka koostuu monista alikonsepteista. ISO-standardin 9241-210 (ISO, 2010) mukaan käyttäjäkokemus on henkilön käsityksiä ja vastauksia, jotka johtuvat käytöstä tai odotetusta tuotteesta, järjestelmän tai palvelun käytöstä. Standardi sisältää seuraavat alaluokat käyttäjäkokemukselle: käyttäjän tunteet, uskomukset, mieltymykset, psykologiset ja fyysiset vaatimukset, käyttäytyminen ja tulokset. Nämä muodostuvat joko ennen järjestelmän tai palvelun käyttöä, käytön aikana tai sen jälkeen. Toisaalta ISO-standardin (ISO, 2010) mukaan käyttäjäkokemus koostuu kahdesta osasta, joita ovat käyttäjän kokemukset sekä järjestelmän tai palvelun käytettävyys.

Käyttäjäkokemukselle on määriteltävissä kolme erottuvaa ominaisuutta, joita ovat hyödyllisyys, käytettävyys ja toivottavuus. Chesnutin ja Nicholsin (2014, ss. 10-13) mukaan ratkaisun tulee tarjota käyttäjille hyödyllistä sisältöä ja toimintoja, jotka ovat käyttäjän tarpeiden mukaisia. Käytettävyydellä tarkoitetaan intuitiivisuutta sekä helppoutta. Tuotavuus taas tarkoittaa sitä, että ratkaisun sisältö annetaan käyttäjälle mielenkiintoisella ja miellyttävällä tavalla.

Chesnut ja Nichols (2014, ss. 9-11) jakavat käyttäjäkokemuksen viiteen osaan, joita ovat tietojärjestelmäarkkitehtuuri, sisältöstrategia, vuorovaikutuksen suunnittelu, käytettävyys ja visuaalinen suunnittelu. Tietojärjestelmäarkkitehtuuri pitää sisällään järjestelmän pienemmät moduulit kuten navigoinnin, sisällön organisoinnin, sivun elementtien visuaalisen prioriteetin eli missä asiat fyysisesti sijaitsevat näytöllä sekä vuorovaikutuksen suunnittelun. Sisältöstrategialla tarkoitetaan sitä, että kaikella sisällöllä olisi oltava jonkinlainen tarkoitus ja sen avulla suunnitellaan, luodaan, toimitetaan ja hallitaan sisältöä. Sisältöstrategian tarkoituksena on luoda käyttäjälle käytännöllistä, haluttua ja hyödyllistä sisältöä. Visuaalinen suunnittelu puolestaan tarkoittaa kaikkia visuaalisia ja esteettisiä näkökulmia kuten värejä. Chesnutin ja Nicholsin (2014, s. 11) mukaan visuaalisen suunnittelun merkitys on hyvin korkealla ajateltaessa käyttäjäkokemusta.

Law et al. (2009) tutkivat kyselyn avulla soveltuuko ISO-standardi (ISO, 9241-210) käyttäjäkokemuksen ammattilaisten käsityksiin. Tutkimuksessa toteutetun kyselyn avulla onnistuttiin selvittämään tieteellisiä ja käytännöllisiä näkemyksiä käyttäjäkokemuksesta sekä ymmärtämään käsitteen laajuutta ja luonnetta. Vastaajat olivat samaa mieltä siitä, että käyttäjäkokemus voidaan nähdä dynaamisena, sisällöriippuvaisena ja subjektiivisena kokonaisuutena. Law et al. (2009) esittävätkin, että termi käyttäjäkokemus on määritelty tuotteisiin, järjestelmiin, palveluihin ja kohteisiin, joita henkilö käyttää käyttöliittymän kautta.

Käyttäjälähtöinen suunnittelu (*eng. user-centered design, UCD*) on lähestymistapa tuotekehitykseen, joka ottaa huomioon loppukäyttäjän tekemällä käyttäjä tutkimusta sekä ottamalla ne mukaan iteratiiviseen tuotteen suunnitteluun ja arviointiin. Käyttäjälähtöisen suunnittelun tavoitteena on tuottaa käyttökelpoisia tuotteita, jotka vastaavat käyttäjien tarpeita. (VTT, 2019; Rubin & Chisnell, 2008, ss. 11-13) UCD ei ainoastaan kuvaa tekniikoita, prosesseja, menetelmiä tai menettelyjä vaan se on myös filosofia, joka asettaa käyttäjän prosessin keskipisteeseen (Rubin & Chisnell, 2008, ss. 11-12). Käyttäjälähtöinen suunnittelu ei ole täydellinen metodologia ohjelmistokehitykseen, kun mietitään koko projektin elinkaarta ja aktiviteetteja. Tarkoituksena on integroida käyttäjälähtöinen suunnittelu aliprosessina olemassa olevalle ohjelmistokehitysmenetelmälle kuten ketterälle projektimenetelmälle. (Gulliksen et al. 2003)

Käyttäjälähtöinen suunnittelu käsittelee hyvin pitkälle asioita, joilla parannetaan käytettävyyttä. Tämän takia kirjallisuudessa usein rinnastetaan käytettävyys samaksi prosessiksi ja toiminnaksi kuin käyttäjälähtöinen suunnittelu. Käyttäjäkokemus puolestaan on paljon laajempi käsite ja konsepti kuin käytettävyys ja se kattaa kaikki loppukäyttäjän vuorovaikutuksen yrityksen, yrityksen palveluiden ja tuotteiden kanssa. (Nielsen Norman Group, 2009; Göransson, 2004, s. 36)

ISO 13407-standardin (ISO 13407, 1999) mukaan käyttäjälähtöinen suunnittelu on lähestymistapa vuorovaikutteiseen järjestelmäkehitykseen, joka keskittyy erityisesti järjestelmien tai sovellusten helppokäyttöisyyteen. Hyödyntämällä käyttäjälähtöistä suunnittelua päädytään tuotteisiin:

- Jotka auttavat käyttäjää saavuttamaan tavoitteet
- Joita ihmiset haluavat käyttää
- Joita ihmiset ymmärtävät, miten tulee käyttää
- Joita voidaan käyttää onnistuneesti

- Jotka tyydyttävät käyttäjät
- Joka estää käyttäjää tekemästä virheitä

Käyttäjälähtöistä suunnittelua ei tule nähdä vain kerran tapahtuvana aktiviteettina, joka tehdään ennen tuotteen julkaisua vaan ennemminkin aktiviteetteina, joita toteutetaan koko tuotekehityksen elinkaaren aikana. (Mao et al. 2005) On olemassa monia erilaisia tasoja käyttäjälähtöisessä suunnittelussa, jotka täydentävät toisiaan jokaisessa projektissa. Käytettävyyden ei tule olla erillään pidettävä käsite tuotekehityksestä ja sen kontekstista. Vaiva mikä nähdään käytettävyyden eteen voi investointina tuottaa tulevaisuudessa esimerkiksi, jos jokin uusi julkaisu tai tuoteperheen tuote hyötyy aikaisemmasta vaivasta. (Nielsen 1993, ss. 70-74)

On olemassa monia erilaisia käyttäjälähtöisen suunnittelun metodeja ja käytäntöjä, jotka sopivat erilaisiin tarkoituksiin. Usein metodit ja käytännöt sisältävät iteratiivisia aktiviteetteja kuten tutkimuksia, suunnittelua ja arviointia joihin loppukäyttäjä osallistuu. (ISO, 2010) ISO-standardi 13407 (ISO 13407, 1999) määrittelee yhden lähestymistavan käyttäjälähtöiseen suunnitteluun. Standardi kuvaa käyttäjälähtöisen suunnittelun siten, että sille on ominaista aktiivinen osallistaminen, selkeä ymmärrys käyttäjien sekä tehtävien vaatimuksista, asianmukainen funktioiden allokointi käyttäjien ja teknologian välillä, suunnitteluratkaisun iterointi sekä monialainen design.

Käyttäjälähtöinen suunnittelu on iteratiivinen tuotekehitysprosessi, jonka aikana loppukäyttäjien palautteet tuodaan mukaan suunnitteluun eri kehitysprosessin vaiheissa (Mao et al. 2005). Gulliksen et al. (2003) kuitenkin kritisoivat ISO-standardin määritelmää ja ajatellaankin, että se ei ole riittävä ylläpitämään käyttäjälähtöistä suunnittelua projekteissa tai ylipäättänsä organisaatiossa. Tämän takia käyttäjälähtöisen suunnittelun prosessia tulisi määritellä yksityiskohtaisemmin, jotta voidaan välttää ongelmia. Gulliksen et al. (2003) ovat määrittäneet 12 tärkeintä periaatetta käyttäjälähtöiselle suunnittelulle. Periaatteet on selitetty auki taulukossa 8.

Taulukko 8. Periaatteet käyttäjälähtöiselle suunnittelulle (Gulliksen et al. 2003)

Periaate	Selitys
<i>Keskitytään käyttäjään</i>	Periaatteen tavoitteet ovat: työskentelykehys tai käyttökonteksti, käyttäjätavoitteet, sekä tehtävien ja tarpeiden tulisi ohjata kehitystä alkuvaiheessa.

<i>Aktiivinen käyttäjien osallistaminen</i>	Käyttäjät, jotka edustavat käyttökohdetta tulisi aktiivisesti osallistua kehitykseen aikaisessa vaiheessa ja jatkuvasti läpi kehitysprosessin ja läpi systeemin elinkaaren.
<i>Evoluutionaaristen systeemien kehitys</i>	Kehityksen tulisi olla sekä iteratiivista että inkrementaalista.
<i>Yksinkertaiset suunnitteluesitykset</i>	Järjestelmän suunnittelu tulisi esittää niin, että se on helpposti ymmärrettävissä käyttäjien ja sidosryhmien keskuudessa.
<i>Prototyyppien muodostaminen</i>	Muodostetaan mahdollisimman aikaisessa vaiheessa ja jatketaan muodostamista prosessin edetessä. Prototyyppejä tulisi käyttää visualisoimaan ja evaluoimaan ideoita ja suunnitteluratkaisuja yhdessä loppukäyttäjien kanssa.
<i>Arvioi käyttöä kontekstissa</i>	Alleviivattujen käytettävyystavoitteiden ja suunnittelukriteerien tulisi kontrolloida kehitystä.
<i>Ammattimainen suhtautuminen</i>	Kehitysprosessi tulisi suorittaa tehokkaiden monitieteellisten tiimien toimesta.
<i>Täsmälliset ja tiedostetut suunnittelutehtävät</i>	Kehitysprosessin tulisi sisältää täsmällisiä ja ennalta tiedostettuja suunnitteluaktiviteetteja.
<i>Käytettävyysohjaaja</i>	Käyttäjäekspertit tulisi ottaa mukaan prosessiin aikaisessa vaiheessa ja jatkuvasti läpi kehityssyklin.
<i>Holistinen suunnittelu</i>	Kaikkia näkökulmia, jotka vaikuttavat tulevaisuuden käyttötilanteisiin tulisi kehittää rinnakkain.
<i>Prosessin kustomointi</i>	UCD prosessi tulisi spesifioida, adoptoida ja/tai implementoida paikallisesti jokaisessa organisaatiossa.
<i>Käyttäjäkeskeinen asenne tulisi aina rakentaa ja tuoda esille</i>	Kaikki ihmiset, jotka ovat mukana tulisi olla tietoisia ja sitoutua käyttäjien ja käytettävyyden tärkeyteen.

Tulee kuitenkin muistaa, että käyttäjäkeskeinen suunnitteluprosessi ei ole täydellinen prosessimalli ohjelmistokehitykseen vaan se tulee integroida johonkin ohjelmistokehityksen metodiin. Käyttäjäkeskeinen suunnitteluprosessi sekä ohjelmistokehityksen metodi luovat yhdessä kokonaisen elinkaaren projektille.

Ihmislähtöinen suunnitteluprosessi

ISO-standardi 13407 on määrittänyt käyttäjäkeskeisen suunnittelun käsitteen jo monia vuosia sitten. Vuosien aikana on kuitenkin huomattu, että käyttäjäkeskeinen suunnittelu ei terminä vastaa täysin siihen, miten asioita tulisi nykyään suunnitella tehokkaasti. ISO-standardi 9241-210 määrittelee käyttäjäkeskeisen suunnittelun nykyisin ihmiskes-

keiseksi. Ihmiskeskeisen suunnittelun tarkoituksena on laajentaa käyttäjäkeskeistä näkökulmaa myös kaikkiin muihin sidosryhmiin. Ihmiskeskeinen suunnittelu eli HCD (*eng. Human-Centered Design*) ottaakin huomioon kaikki tuotteen käyttäjät sekä sidosryhmät. Sidosryhmillä tarkoitetaan kaikkia, joihin järjestelmä voi vaikuttaa, mutta eivät välttämättä itse käytä järjestelmää. (Norman, 2013; Heinilä et al. 2005) Ihmiskeskeinen suunnittelu täydentää käyttäjäkeskeisen suunnittelun periaatteita.

Ihmislähtöinen suunnittelu eli HCD on järjestelmien kehitykseen tarkoitettu lähestymistapa (ISO, 2010). HCD:n tarkoituksena on käyttäjäkeskeisen suunnittelun tapaan tehdä järjestelmästä mahdollisimman hyödyllinen ja käytettävä. Ihmislähtöinen suunnittelu ottaa huomioon järjestelmän sidosryhmien edustajat ja heidän tarpeensa toisin kuin käyttäjälähtöinen suunnittelu, jossa keskitytään erityisesti loppukäyttäjiiin. (Norman, 2013; ISO, 2010) Ihmislähtöinen suunnittelu pyrkii mallintamaan asioita oikeiden käyttäjien ja sidosryhmien avulla (Norman, 2013). Menetelmän tarkoituksena on ajatella käyttäjät ja sidosryhmät oikeina ihmisinä eikä vain abstrakteina käsitteinä. Kun osataan ajatella sidosryhmät oikeina henkilöinä, tulee myös löytää oikea ja ratkaistava ongelma. Jokaista ongelmaa ei kannata tai edes pystytäkään ratkaisemaan ja siksi tulee miettiä, millainen ongelma voidaan ratkaista. (Norman, 2013; ISO, 2010)

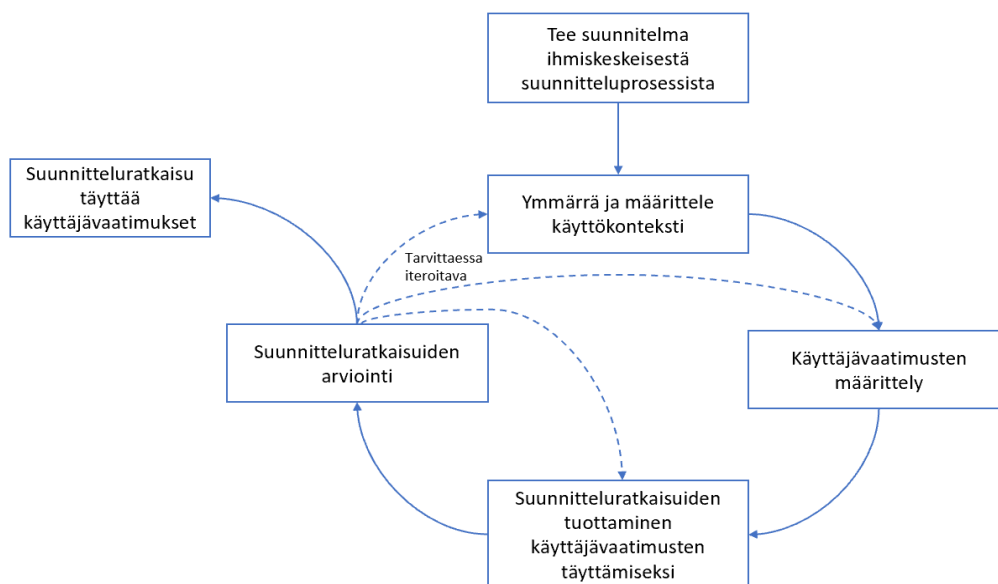
ISO 9241-210 (ISO, 2010) standardi määrittää, että ihmislähtöinen suunnitteluprosessi on kehitetty auttamaan henkilöitä, joiden vastuulla on iteratiivinen järjestelmien suunnittelu ja kehittäminen. Suurimmat hyödyt, joita ihmislähtöinen suunnitteluprosessi mahdollistaa ovat taloudellisia ja sosiaalisia hyötyjä, joita järjestelmän käyttäjät, organisaation työntekijät ja alihankkijat voivat saavuttaa. Ihmislähtöinen suunnittelu mahdollistaa sen, että järjestelmät ovat käytettävämpiä sekä antavat käyttäjille paremman käyttökokemuksen. ISO-standardi (ISO, 2010) määrittelee hyötyjä, joita voidaan saavuttaa hyödyntämällä ihmislähtöistä suunnitteluprosessia:

- Käyttäjäkokemus paranee
- Käyttäjät ymmärtävät järjestelmää helpommin ja osaavat käyttää sitä ilman jatkuvaa koulutusta ja tukemista
- Tuottavuus kasvaa erityisesti käyttäjillä yhdessä operationaalisen tehokkuuden kanssa
- Kun järjestelmä on käytettävä aiheuttaa se vähemmän epämukavuutta käyttäjille
- Funktionaalisten vaatimusten tunnistaminen ja määrittäminen helpottuu
- Mahdollistaa projektin onnistumisen aikataulun ja budjetin suhteen

- Pienentää riskiä epäonnistua vaatimusten toteuttamisessa ja tekee järjestelmästä sellaisen, jota myös käytetään
- Brändin imago paranee ja saavutetaan parempaa kilpailuetua

Ihmiskeskeinen suunnittelu ottaa huomioon keskeiset käyttäjät ja sidosryhmät aina suunnittelusta lopulliseen toteutukseen asti. Käyttäjäkeskeisessä suunnittelussa (UCD) ajatellaan, että käyttäjä on asiantuntija ja järjestelmiä kehitetään heitä varten. (ISO, 2010; Beyer et al. 2010) Tämä sama ajattelutapa sisältyy myös ihmiskeskeiseen suunnitteluun. Käyttäjien rooli molemmissa suunnittelumenetelmissä on hyvin tärkeä ja ajatellaankin, että käyttäjä on aina viimeinen auktoriteetti sanomaan, onko järjestelmä hyvä vai ei. (Beyer et al. 2010)

Ihmislähtöinen suunnittelu voidaan nähdä nelivaiheisena prosessina, joka Normanin (2013) mukaan koostuu havainnoinnista, ideoinnista, prototyyppien tekemisestä sekä testauksesta. ISO standardin (ISO, 2010) mukaan ihmislähtöinen suunnitteluprosessi koostuu myös neljästä vaiheesta, jotka on nimetty hieman eri tavalla. Näitä vaiheita ovat käyttökontekstin ymmärtäminen ja määrittäminen, käyttäjävaatimusten määrittäminen, suunnitteluratkaisuiden toteutus käyttäjävaatimusten täyttämiseksi sekä suunnitteluratkaisuiden arviointi vaatimusten suhteen. Kuvassa 7 on mallinnettu ISO-standardin mukainen ihmislähtöinen suunnitteluprosessi.



Kuva 7. ISO 9241-210 standardin mukainen ihmislähtöinen suunnitteluprosessi (ISO 2010)

Kun lähdetään suunnittelemaan ohjelmistoa, tulee aluksi tehdä suunnitelma ihmiskeskeisestä suunnitteluprosessista. Tässä vaiheessa on hyvä määrittää millaisia rajoitteita

ja vaatimuksia ohjelmistolle on. (Beyer et al. 2010; ISO, 2010) Suunnitteluprosessin kannalta tulee tietää, kuinka suuri järjestelmä on, kuinka paljon käyttäjiä järjestelmällä on, onko olemassa muita järjestelmiä, joiden kanssa järjestelmän tulisi toimia yhteen tai onko jotain turvallisuusaspekteja, jotka tulee ottaa huomioon (ISO, 2010). Tämän lisäksi on hyvä miettiä, liittyykö järjestelmän huonoon käytettävyyteen jotain riskejä sekä tärkeää on määrittää kehitysympäristö. Kehitysympäristöllä tarkoitetaan sitä, kuinka laaja projekti on, millainen budjetti ja aikataulu on, mitä teknologioita käytetään ja niin edelleen. (Collier, 2011; ISO, 2010)

Prosessin toisessa vaiheessa ymmärretään ja määritetään käyttökonteksti. Uuden järjestelmän käyttökontekstia ymmärrettäessä tulee myös tarkastella olemassa olevan ratkaisun käyttökontekstia. (Gulliksen et al. 2003) Koska käyttökontekstilla tarkoitetaan järjestelmän käyttäjien ja tehtävien ominaisuuksia sekä teknistä ja fyysistä ympäristöä, jossa järjestelmää käytetään olemassa olevaan ratkaisuun tutustuminen auttaisi ymmärtämään mitä heikkouksia siitä on tunnistettavissa ja mihin erityisesti halutaan parannusta uuden järjestelmän myötä. (Gulliksen et al. 2003; ISO, 2010) Näitä tekijöitä voidaan pitää lähtökohtana uuden järjestelmän suunnittelussa. ISO-standardin (ISO, 2010) mukaan tulee käyttökontekstin kuvauksesta tuottaa dokumentti, joka ottaa huomioon olemassa olevan ja uuden ratkaisun käyttökontekstin.

Käyttökontekstin kuvaaminen tulee ISO-standardin (ISO, 2010) mukaan mallintaa neljän osa-alueen avulla: käyttäjät ja muut sidosryhmät, käyttäjien ja käyttäjäryhmien ominaisuudet, käyttäjien tavoitteet ja tehtävät sekä järjestelmän ympäristö. Käyttäjien ja muiden sidosryhmien kannalta tulee tunnistaa kaikki sidosryhmät sekä mallinnetaan ryhmien välisiä suhteita. Tässä vaiheessa on myös hyvä miettiä millaisia tavoitteita käyttäjillä ja sidosryhmän edustajilla on järjestelmältä sekä onko tunnistettavissa jotain rajoituksia. Taulukossa 9 on määritelty käyttökontekstin kuvaamisen osa-alueet ja niiden määritelmät.

Taulukko 9 Käyttökontekstin kuvaamisen osa-alueet (ISO 2010)

Osa-alue	Määritelmä
<i>Käyttäjät ja sidosryhmät</i>	Tunnistettava sidosryhmät ja niiden väliset suhteet. Tavoitteet ja rajoitteet, joita sidosryhmien edustajilta on tunnistettavissa
<i>Käyttäjien ja käyttäjäryhmien ominaisuudet</i>	Tunnistetaan käyttäjien ja käyttäjäryhmien tietämys, taidot, kokemus, koulutus, työtehtävät, mieltymykset ja

	muut asiat, jotka vaikuttavat järjestelmän käytön kanalta oleellisilta.
<i>Käyttäjien tavoitteet ja tehtävät</i>	Tunnistetaan käyttäjien tavoitteet ja kuvataan ne. Mietitään kuinka usein ja miten käyttäjät suorittavat vaadittavia tehtäviä järjestelmällä. Hyvä tarkastella kuinka kauan tehtävien tekemiseen menee ja onko tehtävien suorittamisen välillä jotain riippuvuuksia. Tässä vaiheessa tulee myös tunnistaa ja kuvata jos tehtäviä on mahdollista suorittaa väärin ja mitä tällaisesta väärinkäytöstä seuraa.
<i>Järjestelmän ympäristö</i>	Tunnistettava järjestelmän ympäristö, joka sisältää eri laitteet ja ohjelmistot. Samalla järjestelmän sosiaalinen, fyysinen ja kulttuurinen ympäristö tulee kuvailla.

Kun järjestelmän käyttökonteksti on määritelty, voidaan siirtyä käyttäjävaatimusten määrittämiseen. Käyttäjävaatimuksien avulla ilmaistaan sitä, mitä käyttäjän tulisi saavuttaa järjestelmän käytöllä. (Gulliksen et al. 2003) Käyttäjätarpeilla ei kuitenkaan haluta ilmaista sitä, miten käyttäjän tulee toimia. Ihmislähtöiselle suunnittelulle on tyypillistä, että käyttäjätarpeita ja -vaatimuksia kuvataan täsmällisesti ja liitetään ne yhteen kokonaisuudeksi käyttökontekstiin ja liiketoiminnallisten tavoitteiden kanssa. (Heinilä et al. 2005; Gulliksen et al. 2003) ISO-standardin mukaan (ISO, 2010) kun määritetään käyttäjävaatimuksia, tulisi määritelmät tehdä tiettyjen näkökulmien kautta:

- Suunniteltu käyttökonteksti
- Vaatimukset, jotka on johdettu käyttökontekstista ja käyttäjätarpeista
- Vaatimukset, jotka on johdettu käyttöliittymästandardeista, -ohjeistuksista ja -tietämyksestä
- Vaatimukset ja tavoitteet, jotka liittyvät käytettävyyteen
- Mitattavat suoritus- ja kyvykkyyksikriteerit määritellyssä käyttökontekstissa
- Järjestelmävaatimukset, jotka pohjautuvat organisaation tarpeisiin ja joilla nähdään olevan vaikutusta järjestelmän käyttäjiin

Suunnitteluprosessin kolmannessa vaiheessa tuotetaan suunnitteluratkaisuja käyttäjävaatimusten täyttämiseksi. Tässä vaiheessa syntyy usein uusia vaatimuksia, kun käyttäjä pääsee arvioimaan jonkinlaista versiota järjestelmästä. (Cooper et al. 2007; ISO, 2010; Stickdorn et al. 2018) Kun suunnitteluratkaisuja tuotetaan, tulisi siinä huomioida:

- Ratkaisuiden konkretisointi prototyyppien, mallien tai käyttötapausten kautta
- Ratkaisuiden korjaaminen kerätyn palautteen tai käyttäjälähtöisyyden arvioinnin perusteella
- Ratkaisut jaettava projektitiimin kesken
- Erilaiset käyttäjätehtävät, käyttäjien ja järjestelmän vuorovaikutus sekä käyttöliittymän suunnittelu

Prosessin viimeisessä vaiheessa arvioidaan suunnitteluratkaisuja vaatimusten suhteen. Arvioinnin avulla voidaan kerätä tarpeellista tietoa käyttäjätarpeista sekä miettiä, onko kaikki asetetut vaatimukset saavutettu. Samalla voidaan kerätä palautetta suunnitelman hyvistä ja huonoista puolista erityisesti käyttäjän näkökulmasta. (Gulliksen et al. 2003; Heinilä et al. 2005, ISO, 2010)

Arviointia voi tapahtua projektin eri vaiheissa esimerkiksi laadittujen prototyyppien avulla. Prototyyppejä testattaessa tulee käyttäjälle antaa joitain ennalta määriteltyjä tehtäviä, joita hänen tulee suorittaa. (Collier, 2011; ISO, 2010) Tällöin testauksen aikana saadaan esille paljon enemmän tekijöitä, jotka toimivat ja asioita, jotka kaipaavat vielä kehitystä. Pelkkä ominaisuuksien demonstroiminen ei tuota samanlaista arvoa projektille.

ISO-standardi (2010) listaa erilaisia esimerkkejä ihmislähtöisen suunnitteluprosessin tuotoksista. Nämä esimerkit on listattu taulukossa 10.

Taulukko 10. Ihmislähtöisen suunnitteluprosessin tuotoksia (ISO 2010)

Prosessivaihe	Tuotos
<i>Käyttökontekstin määrittäminen</i>	Käyttökontekstin kuvaus
<i>Käyttäjävaatimusten määrittäminen</i>	Käyttäjätarpeiden kuvaus Käyttötapausten määrittäminen Käyttäjävaatimusten määrittäminen
<i>Suunnitteluratkaisuiden tuottaminen</i>	Käyttöliittymän määrittäminen Käyttäjien vuorovaikutuksen määrittäminen Käyttöliittymän implementointi
<i>Ratkaisuiden arviointi</i>	Arvioinnin tulos

Ihmislähtöiselle suunnitteluprosessille on tunnistettavissa myös monia erilaisia haasteita. Ongelmana voidaan pitää sitä, kuinka käyttökonteksti voi vaihdella eri käyttäjäryh-

mien tai toimintojen välillä. (Gulliksen et al. 2003) Koska erilaisia käyttäjäryhmiä järjestelmälle on paljon, on jokaisella myös hieman erilaiset tarpeet järjestelmälle. Suunnitteluprosessin aikana tulisi ottaa huomioon kaikkein käyttäjäryhmien tarpeet. (ISO, 2010) On kuitenkin huomattava, että joskus käyttäjäryhmien välillä tarpeet voivat erota toisistaan niin paljon, että ne ovat toisiaan vastaan. Sekä ihmislähtöiselle että käyttäjälähtöiselle suunnittelulle on olennaista se, että projektin alussa kerätyt järjestelmän vaatimukset vaativat uudelleen määrittelyä prosessin edetessä. Usein uusia vaatimuksia nousee esille, kun järjestelmästä tehdään konkreettinen versio, jota voidaan myös testata. (ISO, 2010; Gulliksen et al. 2013)

4. PALVELUMUOTOILUN HYÖDYNTÄMINEN KÄYTTÄJÄLÄHTÖISESSÄ SUUNNITTELUSSA

Tässä luvussa syvennyttään palvelumuotoiluun sekä erilaisiin palvelumuotoilumetodeihin, joilla voidaan tukea käyttäjälähtöistä suunnittelua. Luvun tarkoituksena on antaa kattava kuva siitä, mitä palvelumuotoilulla tarkoitetaan ja millaisia palvelumuotoilumetodeja on tunnistettavissa kirjallisuudesta.

4.1 Mitä palvelumuotoilu on?

Palvelumuotoilua voidaan hyödyntää laajasti organisaation eri toiminnoissa. Se yhdistää asiakkaat liiketoimintaan ja organisaatioon. (Reason et al. 2015) Palvelumuotoilu tarjoaa perspektiivin, metodeja ja työkaluja, jotka mahdollistavat organisaation liiketoiminnallisten mahdollisuuksien tavoitteiden asettamisen sekä tavan käsitellä sisäisiä ja ulkoisia haasteita. (Teixeira et al. 2012)

Reason et al. (2015) mukaan määrittämällä vastaukset kolmeen eri peruskysymykseen pystytään ymmärtämään ja käsittelemään strategisia aloitteita sekä operationaalisia haasteita. Nämä kolme kysymystä ja lähestymistapaa ovat:

1. Mitä tämä tekee meidän nykyisille ja tuleville asiakkaille?
2. Kuinka tämä vaikuttaa meidän liiketoimintaamme?
3. Mitkä organisaation kyvyt sekä resurssit ovat tarpeellisia, kun aloitteeseen vastataan ja sitä viedään eteenpäin?

Palvelumuotoilu nähdään usein tasapainona asiakkaiden, organisaation sekä liiketoiminnan välillä. Teixeira et al. (2012) mukaan asiakkaat luovat yhteistyössä palvelun tarjoajan kanssa hyvin uniikin kokemuksen. Tämä kokemus muodostuu erilaisten vuorovaikutustilanteiden kautta eri kosketuspisteissä. Reason et al. (2015) korostavat myös sitä, että palvelumuotoilun tarkoituksena on suunnitella palveluita ihmisten kanssa eikä vai ihmisille. Palvelumuotoilu voidaankin nähdä hyvin iteratiivisena kokonaisuutena, joka osallistaa eri tahoja, jotta saavutetaan paras mahdollinen lopputulos.

Stickdorm et al. (2018) mukaan palvelumuotoilun määritelmä ja periaatteet joihin käsite voidaan jakaa muuttuvat jatkuvasti. Kirjoittajat ovat jakaneet vuonna 2010 palvelumuotoilun viiteen periaatteeseen, jotka on esitelty taulukossa 11.

Taulukko 11. *Palvelumuotoilun periaatteet vuonna 2010 (Stickdorm et al. 2018)*

Periaate	Määritelmä
<i>Käyttäjäkeskeinen</i>	Palvelu tulisi kokea asiakkaan silmin
<i>Yhteistyö (eng. co-operative)</i>	Kaikki sidosryhmät tulisi ottaa mukaan palvelumuotoilu-prosessiin
<i>Sekvensoitava</i>	Palvelu tulisi visualisoida toisiin liittyvien toimien sarjana
<i>Osoitettavissa oleva tai todistettavissa oleva</i>	Aineettomat palvelut tulisi visualisoida fyysisinä esineinä
<i>Kokonaisvaltainen</i>	Palvelun koko ympäristö on otettava huomioon

Stickdorm et al. (2018) huomasivat kuitenkin, että määritelmät kaipaivat päivitystä sillä ne eivät myöhemmin kuvanneet enää tarpeeksi tarkasti palvelumuotoilua ja sen ominaisuuksia. Tekijät ovatkin määritelleet uudet kuusi periaatetta, jotka pätevät tänä päivänä palvelumuotoiluun. Nämä määritelmät on esitetty taulukossa 12. Käyttäjäkeskeisyys on vaihtunut ihmislähtöisyydeksi ja palvelumuotoilulla pyritään mallintamaan enemmän todellisia tilanteita sekä ottamaan suunnitteluprosessiin aktiivisemmin mukana kaikki sidosryhmät ja niiden jäsenet. Uusien periaatteiden kautta painotetaan myös enemmän tutkimusta ja kokeellista lähestymistapaa sekä lopputuloksena saavutettavan palvelun tulisi vastata kaikkien sidosryhmien tarpeisiin, ei ainoastaan käyttäjien tarpeisiin.

Keskeisintä periaatteiden muuttumisessa on se, että ennen palvelumuotoilu ajateltiin käyttäjäkeskeiseksi tekemiseksi. Tällä tarkoitetaan sitä, että palvelu tulisi kokea asiakkaan silmin. Nykyisin kuitenkin käyttäjäkeskeisyys ei riitä kattamaan kaikkea mitä palvelumuotoilulla halutaan saavuttaa ja siksi nykyinen uusi muuttunut periaate on ihmiskeskeisyys. Ihmiskeskeisyydellä painotetaan sitä, että palvelu tulisi kokea kaikkien niiden ihmisten silmin, joihin palvelu vaikuttaa. (Stickdorm et al. 2018; Miller, 2015) Näitä ihmisiä voivat olla myös muut kuin käyttäjät ja palvelu voikin vaikuttaa henkilöihin, jotka eivät ole järjestelmän kanssa tekemisissä.

Taulukko 12. *Palvelumuotoilun muuttuneet periaatteet vuonna 2017 (Stickdorm et al. 2018)*

Periaate	Määritelmä
<i>Ihmiskeskeinen</i>	Palvelu tulisi kokea kaikkien ihmisten silmin, joihin palvelu vaikuttaa.
<i>Yhteistyö (eng. collaborative)</i>	Kaikkien sidosryhmien tulee olla aktiivisesti mukana palvelun suunnitteluprosessissa ja sitoutettuna siihen.
<i>Iteratiivinen</i>	Palvelumuotoilu on tutkiva, mukautuva ja kokeellinen lähestymistapa. Iterointikierrosten kautta saavutetaan toteutus.
<i>Sekvensoitava</i>	Palvelu tulisi visualisoida toisiin liittyvien toimien sarjana
<i>Todellinen</i>	Tarpeet tulisi tutkia ja tunnistaa realistisissa tilanteissa, ideat muodostuvat prototyypeiksi todellisissa tilanteissa ja aineettomat arvot todistettava fyysisenä tai digitaalisena todellisuutena.
<i>Kokonaisvaltainen</i>	Palveluiden tulisi vastata kaikkien sidosryhmien tarpeisiin.

Jotta palvelumuotoilusta saadaan onnistunutta ja tehokasta tulisi Stickdorm et al. (2018) mukaan ottaa kokonaisvaltaisesti jokainen periaate ja näkökulma huomioon. Palvelumuotoilu voidaan nähdä niin ajattelutapana kuin prosessina tai isona määränä käteviä työkaluja. Kun tarkastellaan palvelumuotoilua prosessinäkökulmasta, tähtää se innovatiivisiin ja elegantteihin ratkaisuihin. On myös hyvä tiedostaa, että palvelumuotoilun prosessi koostuu iteratiivisista tutkimus- ja kehityssykleistä. (Lawrence et al. 2018)

Palvelumuotoiluun on kehitetty vuosien aikana lukuisia erilaisia työkaluja, joiden tarkoituksena on edistää yhteistyötä ja ymmärryksen luomista (Stickdorm et al. 2018). Useat menetelmät on tarkoitettu herättämään keskustelua ja niiden avulla on osallistujien helpompi jakaa mielipiteitä sekä löytää yhteisiä keinoja, ratkaisuja tai menetelmiä eri osapuolten välillä. Palvelumuotoilun avulla voidaan tehostaa ketterän ohjelmistoprojektin vaatimusmäärittelyä. Erilaisia palvelumetodeja hyödyntämällä voidaan saada tarkempia määritelmiä siitä, mitä eri sidosryhmät haluavat uuden järjestelmän käytöllä saavuttaa. Samalla palvelumuotoilun menetilat luovat ymmärrystä sekä tukea prosessin eri vaiheissa, jotta voidaan saavuttaa vaaditut tavoitteet. Palvelumuotoilu tarjoaakin erilaisia työskentely- ja ajattelutapoja sekä työkaluja, joilla voidaan tukea vaatimusmäärittelyä. (Stickdorm et al. 2018)

4.2 Vaatimusten mallintaminen palvelumuotoilumetodeja hyödyntäen

Tässä kappaleessa esitellään erilaisia palvelumuotoilumetodeja, joita voidaan hyödyntää toiminnallisten vaatimusten mallintamisessa. Nämä menetelmät ovat tunnettuja palvelumuotoilumetodeja, joista osaa on kirjallisuuden perusteella sovellettu jo jollain tasolla vaatimusmäärittelyssä. Menetelmien valinta perustuu kirjallisuuskatsauksen perusteella karttuneeseen tietoon siitä, millainen toiminnallisten vaatimusten määrittämisen prosessimallin tulee olla ja mitä onnistunut vaatimusmäärittely sisältää.

Haastattelut

Yksi perinteisimmistä palvelumuotoilun keinoista ja usein vaatimusmäärittelyssä käytetyistä tekniikoista on haastattelut. Haastattelut ovat ihmispohjaisia toimintoja, jotka ovat luonteeltaan epävirallisia. Haastatteluiden tehokkuus riippuu suuresti osallistujien välisestä vuorovaikutuksen laadusta. Haastattelut tarjoavat tehokkaan tavan kerätä suuria määriä tietoja nopeasti. Haastattelijan taito vaikuttaa paljon siihen kuinka hyödyllistä kerätty tieto on. (Stickdorn & Schneider, 2011; Stickdorn et al. 2018; Lawrence et al. 2018) Haastatteluita on olemassa kolmea erilaista tyyppiä, joita ovat strukturoitu, strukturoimaton ja puolistrukturoitu haastattelu. Strukturoimattomat haastattelut ovat luonteeltaan keskustelevia, jolloin haastattelijalla on valvoo keskustelujen suuntaa vain hyvin vähän. Strukturoimattomat haastattelut eivät noudata ennalta määrättyä esityslistaa tai kysymysluetteloa ja siksi onkin vaarana, että jotkin aihealueet jätetään kokonaan huomiotta tai keskitytään liian yksityiskohtaisesti johonkin aihealueeseen. (Carrizo et al. 2008; Stickdorn et al. 2018)

Strukturoidussa haastattelussa haastattelut suoritetaan käyttämällä ennalta määrättyä joukkoa kysymyksiä jonkin tietyn tiedon keräämiseen. (Zowghi & Coulin, 2005) Strukturoitujen haastatteluiden menestys riippuu siitä, tietääkö haastattelijalla, mitkä ovat oikeat kysymykset aihealueeseen liittyen ja milloin niitä pitäisi kysyä. (Zowghi & Coulin, 2005; Stickdorn et al. 2018) Samalla on myös tärkeää tietää, keneltä näitä kysymyksiä kysytään eli kenen tulisi tietää vastaus määriteltäviin kysymyksiin. Strukturoiduille haastatteluille on olemassa erilaisia malleja, joiden avulla voidaan jäsentellä haastattelussa esitettäviä kysymyksiä. (Zowghi & Coulin, 2005)

Puolistrukturoitu haastattelu on yhdistelmä strukturoidusta ja strukturoimattomasta menetelmästä (Zowghi & Coulin, 2005; Stickdorn et al. 2018). Osa kysymyksistä on valmisteltu etukäteen, mutta haastattelija voi keskustella esille nousseista asioista ja esittää uusia, erilaisia kysymyksiä haastateltavalle. Puolistrukturoitu haastattelu voisi olla paras valinta haastattelumenetelmistä vaatimusmäärittelyprosessiin. (Stickdorn et al. 2018) Aikaisemmin esitelty toiminnallisen vaatimusmäärittelyn prosessimalli ohjaa, mitä tietoa halutaan kussakin prosessin vaiheessa saada ja tämän perusteella voidaan laatia haastattelukysymysten runko, jota hyödynnetään puolistrukturoidussa haastattelussa (Stickdorn et al. 2018; Hormess et al. 2018).

Oikein toteutettuna haastattelut ovat hyvä tapa kerätä tärkeää tietoa (Stickdorn et al. 2018). Jotta haastattelut ovat onnistuneita tulee niissä noudattaa haastatteluiden peruseriaatteita. Haastatteluissa kaikki lähtee liikkeelle luottamuksesta. Haastateltavan tulee tuntee, että haastattelija on kiinnostunut hänen vastauksistaan ja heidän välille tulee heti alussa luoda luottamusta. (Stickdorn et al. 2018; Hormess et al. 2018) Haastattelijan tulee asennoitua haastatteluihin niin, että he ovat aina oppimassa jotain uutta jokaisessa haastattelutilanteessa. Haastatteluiden tarkoituksena ei ole pelkästään vahvistaa haastattelijan ennakkoluuloja, vaan onnistuneet haastattelut nostavat paljon uusia asioita esille, joista haastattelija voi oppia uutta. (Stickdorn et al. 2018; Hormess et al. 2018; Carrizon et al. 2008) Haastattelussa esitettävien kysymysten tulee olla selkeitä ja niitä esitetään vain yksi kerrallaan. Kysymyksien asettelussa tulee välttää sanoja, joita haastateltava ei ymmärrä ja kysymykset tulee olla avoimia eikä niihin tule olla mahdollisuutta vasta ”kyllä” tai ”ei” vastauksia. Haastattelussa käytettävät kysymykset eivät saa olla johdattelevia vaan niiden pitää antaa tilaa vastaajalle tuoda omia näkemyksiä esille. (Stickdorn et al. 2018) Tämän takia haastatteluissa ei saisi pelätä hiljaisia hetkiä vaan antaa haastateltavalle tilaisuus miettiä hetki ennen vastausta. Stickdornin et al. (2018) mukaan haastatteluiden aikana on erityisen tärkeää kuunnella haastateltavaa. Usein tätä vähätellään eikä tilanteen tullen osata oikeasti antaa haastateltavalle tilaa.

Stickdorn et al. (2018) nostaa esille myös sen, kuinka tärkeää on välillä haastatteluiden aikana haastattelijan esittää haastateltavan antama vastaus omin sanoin. Tämän tekniikan avulla varmistetaan, että haastattelija on ymmärtänyt asiat niin kuin haastateltava on ne halunnut ilmaista.

Haastatteluita voidaan toteuttaa joko yksilö- tai ryhmähaastatteluina. Carrizo et al. (2018) mukaan yksilöhaastatteluilla saadaan parhaita tuloksia aikaan, mutta ryhmähaastatte-

luillakin voidaan päästä hyviin tuloksiin. Ryhmähaastattelut antavat paljon hyviä vastauksia silloin, kun haastateltavat ovat keskenään samaa mieltä. Jos kuitenkin esimerkiksi vaatimuksia kerätään eri käyttäjäryhmiltä, joilla on erilaiset tarpeita, ei ryhmähaastattelu toimi tiedon keräämisessä. Tällöin joko yksilöhaastattelut tai ryhmien pilkkominen pienempiin mahdollistaisi tehokkaamman tiedon keräämisen.

Palvelumuotoilumetodina haastatteluita voidaan toteuttaa joko syvällisinä haastatteluina tai konseptuaalisina haastatteluina. Syvälliset haastattelut (*eng. in-depth interviews*) tulee toteuttaa tärkeiden sidosryhmien kanssa ja niissä pyritään ymmärtämään jotain näkökulmaa tietyistä aiheista. Syvälliset haastattelut voivat auttaa tutkijaa ymmärtämään ja oppimaan enemmän tietyistä oletuksista, kokemuksista, tuotteista, palveluista, prosesseista tai huolista ja asenteista, joita sidosryhmien edustajilla on. Syvällisiä haastatteluita voidaan toteuttaa strukturoituina, puolistrukturoituina tai strukturoimattomana haastatteluna. (Zowghi & Coulin, 2005; Carrizon et al. 2008) Usein kuitenkin puolistrukturoidun haastattelun avulla voidaan kerätä hyvin paljon hyödyllistä dataa. Syvällisissä haastatteluissa kysymykset tulee järjestää tietyllä tavalla. Ensin tulee esittää yleisiä kysymyksiä aiheesta, jonka avulla haastateltava saadaan rentoutumaan ja keskustelemaan aiheesta vapaasti. Tämän jälkeen voidaan siirtyä esittämään tarkkoja ja yksityiskohtaisia kysymyksiä, jotka liittyvät haastattelun aiheeseen. (Stickdorm et al. 2018)

Kontekstuaalisissa haastatteluissa pyritään ymmärtämään jotain ennalta määriteltä asiakas- tai käyttäjäryhmää. Haastatteluiden aikana halutaan saada tietoa käyttäjäryhmän tarpeista, tunteista, odotuksista sekä ympäristöstä, jossa he työskentelevät. Kontekstuaaliset haastattelut paljastava paljon myös virallisia ja epävirallisia verkostoja ja piilossa olevia aikeita, joita on hyödyllistä ymmärtää hahmoteltaessa uutta järjestelmää. Haastattelun aikana haastattelijan tulee pyrkiä esittämään kysymyksiä tietyistä kokemuksista, joita haastateltavilla on ollut. (Zowghi & Coulin, 2005; Carrizon et al. 2008; Stickdorm et al. 2018) Esimerkiksi selvitetään viimeisintä käyttökertaa vanhan järjestelmän kanssa ja pyritään saamaan haastateltava demonstroimaan yksityiskohtia tästä konkreettisesta kokemuksesta. Käyttäjien on yleensä helpompaa artikuloida ja kertoa vaikeuksista ja hyvistä asioista järjestelmän käyttöön liittyen, kun haastattelun aikana mietitään jotain konkreettista käyttökertaa. (Stickdorm et al. 2018)

Kyselyt

Kyselyitä voidaan käyttää vaatimusmäärittelyssä hyvin varhaisessa vaiheessa. Kyselyt voivat koostua avoimista tai suljetuista kysymyksistä tai molemmista. Jotta kyselyt ovat tehokkaita, tulee kysymysten ja kyselyiden käsitteet sekä rajat määrittää. On myös hyvä

ymmärtää niin kyselylomakkeen suunnittelijan kuin kyselyyn osallistuvien henkilöiden rooleja ja näkökulmia kysymyksiä laatiessa. (Zowghi & Coulin, 2005; Stickdorm, 2018)

Kysymykset tulee laatia niin, että niiden avulla ei kerätä suuria määriä tarpeetonta tai merkityksetöntä tietoa. Kyselyiden avulla pystytään keräämään tietoa useilta sidosryhmiltä nopeasti, mutta kyselyiden avulla pystytään saavuttamaan vain tietynlainen vastausten syvyys. (Zowghi & Coulin, 2005) Kyselylomakkeista puuttuu mahdollisuus tarkentaa tai laajentaa jotain tiettyä osa-aluetta kyselyn edetessä eikä niiden avulla voida tarjota kyselyyn osallistuvalla mahdollisuutta selventää tai oikaista väärinkäsitystä. (Stickdorm, 2018)

Koska kyselyt toimivat parhaiten hyvin varhaisessa vaiheessa vaatimusmäärittelyä voidaan ajatella, että parhaiten kyselytutkimukset toimisivat työkaluna vaatimusmäärittelyprosessin esiselvitysvaiheessa. Kyselytutkimuksen avulla voidaan mallintaa esimerkiksi liiketoiminta-aluetta, johon järjestelmää ollaan kehittämässä tai selventää järjestelmän luonnetta tai kehitysympäristöä. Kyselyt eivät kuitenkaan itsessään riitä kattamaan kaikkia esiselvitysvaiheen kriteereitä. (Zowghi & Coulin, 2005) Usein kyselyiden kysymykset voidaan esittää haastatteluna, sillä vaatimusmäärittelyprosessiin osallistuu rajattu määrä henkilöitä ja haastattelun avulla saadaan laajempia vastauksia (Stickdorm, 2018).

Aivorihi

Aivorihi (*eng. Brainstorming*) on prosessi, jossa eri sidosryhmien edustajat osallistuvat epäviralliseen keskusteluun tuottaakseen mahdollisimman monta ideaa nopeasti keskittymättä mihinkään niistä sen syvällisemmin (Stickdorm, 2018). Tällaista menetelmää hyödynnettäessä on tärkeää välttää ideoiden tarkastelemista tai kritisointia. Aivoriihen tarkoituksena ei ole ratkaista tärkeitä kysymyksiä tai tehdä keskeisiä päätöksiä. (Stickdorm, 2018; Hormess et al. 2018) Aivoriihen suurimpana etuna voidaan pitää sitä, että se edistää ajattelun ja ilmaisun vapauttamista ja mahdollistaa uusien sekä innovatiivisten ratkaisuiden löytämisen olemassa oleviin ongelmiin. (Stickdorm, 2018)

Aivorihi onkin erityisesti ryhmäharjoitus, johon sovelletaan hyvin yksinkertaisia sääntöjä auttamaan osallistujia pysymään tuottavina niin, että ilmapiiri pidetään positiivisena ja mitään ideoita ei tule tuomita (Hormess et al. 2018). Jotta aivorihi menetelmänä olisi onnistunut tulee se toteuttaa niin, että jokainen osallistuja on fyysisesti paikalla ja osallistujien välille luodaan rento mutta luottavainen ilmapiiri. Jokaiselle tilaisuudelle on nimetty oma ohjaaja, joka pitää huolen siitä, että osallistujat pystyvät olemaan tuottavia ja

ideoimaan mahdollisimman paljon ilman suuria rajoitteita. (Lawrence et al. 2018; Stickdorn, 2018)

Aivoriihi menetelmänä sopii erityisesti tilanteisiin, joissa halutaan nopeasti saada ryhmän kesken näkemyksiä siitä, mitä muut ajattelevat aiheesta (Stickdorn et al. 2018). On kuitenkin tärkeää miettiä, onko aivoriihi sopiva menetelmä tiettyyn tilanteeseen (Hormess et al. 2018). Jos halutaan tuottaa monipuolisempia ideoita ja antaa enemmän vaikutusvaltaa ryhmän jäsenille, hiljaisemmat menetelmät toimivat paremmin (Stickdorn et al. 2018).

Aivoriihi menetelmänä on hyvin yksinkertainen järjestää sillä Stickdorn et al. (2018) ohjeistavat, että valmistautumiseen tulisi käyttää korkeintaan 5 minuuttia ja itse aivoriihen tulisi kestää 5-15 minuuttia. Tämän lisäksi on varattava aikaa keskustelulle. Hyvänä ohjeena aivoriihen kokoluokalle on 3-30 osallistujaa. Jos tapahtumaan osallistuu alle 3 henkilöä ei sen aikana saada ideoitua tarpeeksi ja tällöin osallistujat voivat kokea suurta painetta omasta suorituksestaan. Toisaalta taas liian isoa ryhmää on hyvin hankala ohjata.

Kohderyhmä

Kohderyhmä (*eng. Focus Group*) on klassinen laadullinen haastattelumenetelmä, jossa tutkija kutsuu koolle ryhmän ihmisiä ja kysyy heiltä kysymyksiä liittyen johonkin tiettyyn tuotteeseen, palveluun, konseptiin, ongelmaan tai prototyyppiin. Kohderyhmän avulla tutkijat pyrkivät ymmärtämään käsityksiä, mielipiteitä, ideoita tai asenteita tietystä aiheesta. Tarkoituksena on luoda sellainen ilmapiiri haastateltaville, että he haluavat puhua vapaasti annetusta aiheesta juuri heidän omasta näkökulmastaan. (Stickdorn et al. 2018)

Kohderyhmille pidettyjä haastatteluja voidaan kuitenkin joissain tilanteissa pitää huonona vaihtoehtona tai niiden ohjaamiseen liitetään riskejä. Haastatteluiden aikana ei käytetä erilaisia ryhmätehtäviä rajoittavina tekijöinä haastatteluiden aikana. Esimerkiksi yhteistyöpajoissa ryhmä muodostaa yhteisen tekemisen ja keskustelun kautta persoonakuvaus ja muita malleja, joita voidaan käyttää suunnittelun myöhemmissä vaiheissa. Kohderyhmien haastattelut keskittyvät vain valittuun aiheeseen ja niiden onnistuminen riippuu ohjaajan tavasta ohjata keskustelua. Vetäjän on tärkeää välttää erilaisia ajattelun vääristymiä tai rajoituksia, joita kohderyhmän sisällä voi syntyä. (Stickdorn et al. 2018)

5 x Miksi

5 kertaa miksi on haastattelumetodi, joka on tehokas tapa perehtyä paremmin ongelmiin ja niiden juurisyihin. Metodin avulla päästään käsiksi ihmisten uskomuksiin ja motivaatioihin. Metodin tarkoituksena on esittää peräkkäin 5 miksi-kysymystä. (Stickdorn & Hormess et al. 2018, ss. 23-24) Esittämällä 5 kertaa miksi-kysymyksen pystytään pureutumaan edellisen vastauksen syihin ja sitä kautta tunnistamaan oikeat juurisyvät vastausten takaa. Tärkeää on, että haastattelija tunnistaa kohdat, joissa tekniikkaa voidaan hyödyntää sekä tunnistaa olemassa olevan tekniikan. (Stickdorn & Schneider 2011, ss. 166-167)

Käyttäjätarinat

Käyttäjätarinoita (*eng. user stores*) käytetään ohjelmistokehityksessä vaatimusten määrittelyyn käyttäjän tai asiakkaan näkökulmasta. Käyttäjätarinoita voidaan hyödyntää suunnitteluprosessin monessa eri vaiheessa kuten ideoinnissa ja ideoiden valinnassa, prototyyppien laatimisessa tai implementoinnin aikana. Yleisesti käyttäjätarinoita suositellaan muodostettavan työpajassa, johon osallistuu projektitiimi, asiakkaan edustajia, joita järjestelmä koskee sekä tuoteomistaja. (Zowghi & Coulin, 2005; Ramussoni, 2010)

Käyttäjätarinoita voidaan kerätä Ramussonin (2010) mukaan tehokkaasti järjestämällä tarinoiden keräämisen työpajan. Työpajan tarkoituksena on koota kehitystiimin jäsenet ja asiakkaan sidosryhmien edustajat yhteen sekä kirjoittaa mahdollisia käyttäjätarinoita järjestelmästä, jonka he haluaisivat toteutettavan. Työpaja tulee järjestää jo ennen ketterän projektin suunnittelua, jotta tiedetään millaisia ominaisuuksia asiakkaat ja sidosryhmän edustajat uskovat haluavansa näkevänsä uudessa järjestelmässä. Tarinoiden keräämisen tarkoituksena on laajentaa näkökulmaa kehitettävästä järjestelmästä. Työpajalla halutaan löytää mahdollisimman monta ominaisuutta, joita asiakkaat arvostaisivat. Tarkoituksena ei ole myöhemmin toteuttaa kaikkia vaatimuksia, mutta tässä vaiheessa on hyvä kerätä kaikki vaatimukset ylös, jotta pystytään kartoittamaan koko kehitettävän järjestelmän kokonaisuus.

Ramussonin (2010) mukaan noudattamalla viittä ohjetta pystytään pitämään hyvät tarinoiden keräämisen työpaja. Näitä ohjeita ovat:

1. Hanki iso avoin huone
2. Piirrä paljon kuvia
3. Kirjoita paljon tarinoita
4. Pidä aivoriihiä yllä kaikista muista tärkeistä asioista
5. Siivoa lista ja tee siitä yksinkertainen sekä helposti ymmärrettävä

Työpaja tulisi pitää isossa avoimessa huoneessa, jossa on tilaa liikkua ja seinille pystytään ripustamaan erilaisia kuvia. (Hughes 2008) Työpajan aikana tulee piirtää mahdollisimman paljon erilaisia kuvia, sillä niiden avulla voidaan ideoida erilaisia asioita, mitä järjestelmältä halutaan. Esimerkiksi persoonien avulla voidaan kuvata ihmisiä, jotka käyttävät järjestelmää ja niiden avulla pystytään ymmärtämään asiakasta paremmin. Vuokaaviot, prosessivirrät ja skenaariot ovat hyviä roolien hahmottamiseen sekä niiden avulla saadaan käsitys siitä, kuinka järjestelmän tulee toimia. Järjestelmäkarttojen ja tietoarkkitehtuurikaavioiden avulla pystytään organisoimaan ja jakamaan työtä. Kuvien avulla muodostetaan kokonaiskuva siitä, miten järjestelmän tulee toimia. Kun kokonaiskuva on hahmotettu, voidaan kuvista alkaa muodostamaan tarinoita. (Ramussoni, 2008; Collier, 2011; Hughes, 2008)

Käyttäjätarinoita muodostettaessa aikaisemmin luodut kuvat otetaan tarkempaan tarkasteluun ja niitä käydään yksitellen läpi asiakkaiden tai sidosryhmien kanssa. Tarinoista pyritään muodostamaan yksinkertaisia, pieniä ja itsenäisiä vaatimuksia, joiden tekeminen vie maksimissaan viisi työpäivää. Liian suuria tarinoita kutsutaan epiikeiksi ja niiden toteuttamiseen menee usein muutaman työviikko. Näitä tarinoita tulee jakaa pienemmiksi, jotta niistä saadaan helpommin toteutettavia vaatimuksia. (Collier, 2011)

Työpajan tarkoituksena on muodostaa mahdollisimman paljon käyttäjätarinoita asiakkaan vaatimuksista. (Collier, 2011) Käyttäjätarinoiden lisäksi työpajassa tulee miettiä samalla myös muita asioita, jotka vaikuttavat projektiin ja jotka huomioon ottamalla projektista saadaan onnistunut. Aikaisemmin luotujen kuvien avulla ei pystytä mallintamaan kaikkea, mitä projektin aikana tulee tehdä. Esimerkiksi datan migraatiot, koulutusmateriaalit, kuormitustestaukset ja niin edelleen. Työpaja on hyvä tilaisuus miettiä myös kaikkia muita projektiin vaikuttavia tekijöitä, jotta projekti on menestyksekkäs. (Ramussoni, 2008; Collier, 2011; Hughes, 2008)

Kun kaikki vaatimukset on saatu listattua käyttäjätarinoiden muodossa, voidaan tätä listaa tarkastella ja siivota. Tässä vaiheessa tulee käydä läpi duplikaatit ja asioita, joita ehkä jostain syystä on unohdettu ottaa huomioon. Samalla käyttäjätarinoita tulee ryhmitellä jollain loogisella tavalla, jotta niitä on myöhemmin helpompi käsitellä. (Ramussoni, 2008)

Yhteistyöpajat

Työpajojen tarkoituksena on yhteistyön avulla muodostaa erilaisia visualisointeja käyttäjien vaatimuksista. Työpajojen aikana usein luodaan oletukseen perustuvia persoonia, järjestelmäkartoja tai käyttäjän sekä asiakkaanpolun malleja. Nämä oletuksiin perustuvat mallinnukset auttavat suunnittelemaan tehokkaasti suunnitteluprosessia ja antavat kuvan siitä, keitä kannattaa suunnitteluun osallistaa, missä vaiheessa ja mitä tietoa heiltä halutaan. Nämä tulokset on ymmärrettävä kehitystyökaluina ja ne voivat olla erittäin arvokkaita ryhmälle yhteisenä lähtökohtana tutkimusprosessin suunnittelulle tai kerätyn tiedon arvioimiseksi ja parantamiseksi. Riskinä voidaan kuitenkin pitää sitä, että näiden mallinnusten avulla vain vahvistetaan oletuksia, eikä välttämättä oteta kaikkia asioita huomioon. (Stickdorn et al. 2018) Tämän takia onkin hyvä, että työpajoihin osallistuu mahdollisimman paljon eri sidosryhmien edustajia. Yhteisen keskustelun myötä pystytään paljastamaan asioita, joita ei välttämättä kaikissa tilanteissa nousisi esille.

Koska yhteistyöpajoihin on tärkeää saada mahdollisimman paljon eri sidosryhmien edustajia, joihin kehitettävä järjestelmä vaikuttaa on tärkeää miettiä, ketä sinne kutsutaan. Työpajojen lopputulos on täysin riippuvainen osallistujien tietämyksestä käsiteltävään aiheeseen. Jos työpajaan kutsuu henkilöitä joihin aihe ei suoraan vaikuta, voidaan heiltä saada hyvin pinnallisia ja abstrakteja tietoja aiheesta. Tulokset saattavat näyttää vakuuttavilta, mutta usein niitä voidaan pitää hyvin puolueellisina. Onnistuneessa työpajassa osallistujat on valittu heidän tietämyksensä perusteella ja työpajan vetäjä on etukäteen suunnitellut, mitä työpajan avulla haluaa saavuttaa ja miten tavoitteisiin päästään. (Stickdorn et al. 2018) Seuraavissa kappaleissa esitellään hieman tarkemmin erilaisia palvelumuotoilun metodeja, joita voidaan hyödyntää yhteistyöpajoissa.

Sidosryhmäkartat ja persoonat

Järjestelmän sidosryhmiä voidaan mallintaa sidosryhmäkartan avulla. Sidosryhmäkartta toimii visuaalisena esityksenä järjestelmän eri sidosryhmistä. Kartan avulla pyritään tunnistamaan tärkeimmät sidosryhmät, määritetään heidän motivaatiot sekä mallinnetaan eri ryhmien välisiä suhteita. Sidosryhmien mallintaminen tapahtuu tehokkaimmillaan työpajassa, jossa laaditaan ensin kattava listaus järjestelmän sidosryhmistä. Tämän jälkeen jokaiselle tunnistetulle sidosryhmälle listataan heidän motivaatiot ja kiinnostuksen kohteet järjestelmää kohden. Kun sekä sidosryhmät että heidän motivaatiot on tunnistettu, voidaan analysoida sidosryhmien sidoksia ja miten ne vaikuttavat toisiinsa. Tämän tarkastelun avulla voidaan esimerkiksi ryhmitellä erilaisia sidosryhmiä yhteisten kiinnostusten ja tarpeiden mukaan. (Stickdorn & Schneider, 2011, ss. 149-151)

Kun sidosryhmät on mallinnettu sidosryhmäkartan avulla, voidaan yksittäisistä sidosryhmien edustajista laatia persoonia. Henkilöhahmolla tulee olla tiettyjä ominaisuuksia, tarpeita, haluja ja motivaatiota, jotka edustavat juuri hänen sidosryhmäänsä. Näihin tekijöihin tulee kehittäjien sekä asiakkaan pystyä samaistumaan, jotta suunnittelusta pystytään tekemään ihmislähtöisempää. Persoonien avulla mallinnetaan esimerkiksi käyttäjäryhmää, joilla on samat kiinnostuksen kohteet, samanlaisia käyttötarpeita tai ovat maantieteellisesti samanlaisia. (Stickdorn, 2018) Usein kuitenkin väestötieteellinen informaatio kuten ikä ja sukupuoli ovat harhaanjohtavia ja niitä tulisi välttää, jotta ei muodosteta minikäänlaisia stereotypioita. Persoonia muodostettaessa voidaan hyödyntää jo olemassa olevia markkinasegmenttejä tai asiakassegmenttejä tai sitten voidaan tarkastella mahdollisia uusia segmenttejä, joilla nähdään jonkinlaista potentiaalia liiketoiminnan kannalta. (Stickdorn, 2018; Stickdorn & Schneider, 2011, ss. 150-155)

Kun luodaan persoonia eri markkinasegmenteistä, tulisi niitä luoda arviolta 3-7 ydinpersoonaa. Jos luodaan enemmän kuin 7 persoonakuvausta on hyvin todennäköistä, että niitä kaikkia ei hyödynnetä. Usein tällainen tilanne johtuu täysin siitä, että unohdetaan kuvausten olemassaolo ja keskitytään vain niihin mitkä muistetaan. (Cohn, 2004; Collier, 2011, ss. 92–95; Lawrence et al. 2018) Persoonakuvauksia luotaessa hyödynnetään usein erilaisia tekniikoita, jotta kuvauksesta saadaan mahdollisimman tarkka ja se mallintaisi oikeita asioita. Stickdornin ja Schneiderin (2011) mukaan olisi hyvä aloittaa yksinkertaisella nopealla oletuksiin perustuvilla luonnoksilla, joita käyttäjät tai kohderyhmän edustajat kirjoittavat itsestään. Tämän jälkeen voidaan pitää yhteistyöpaja, johon osallistuu etulinjan työntekijät sekä muita sidosryhmien edustajia, joita koetaan tärkeiksi persoonia muodostettaessa. Työpajassa luodaan vielä enemmän oletuksiin perustuvia persoonakuvauksia, joita samalla iteroidaan vastaamaan käyttäjätarpeita.

Asiakkaan polku ja järjestelmäkartat

Polkujen mallintamisessa voidaan visualisoida olemassa olevia ominaisuuksia tai niillä voidaan mallintaa suunnitteluvaiheessa olevia vielä toteuttamattomia ominaisuuksia. Polun mallintaminen keskittyy ihmiskokemukseen kuvaamalla joitain tiettyjä tekijöitä tai vaiheita jonkin tietyn asian tekemisessä. (Stickdorn & Schneider 2011) Persoonakuvausten avulla pystyttiin mallintamaan sidosryhmän edustajaa sekä heidän vaatimuksiansa. Asiakkaan polun avulla pyritään mallintamaan sidosryhmän edustajan kokemusta jossain tiettyssä kontekstissa. Asiakkaan polku koostuu vaiheista, joiden avulla visualisoidaan jokin käyttökokemus ja kaikki siihen vaikuttavat tekijät kuten taustaprosessit ja fyysiset tekijät, joita käyttäjä näkee esimerkiksi tietokoneen näytöllä. Malli tarjoaa hyvin selkeän

ja yksityiskohtaisen kuvauksen järjestelmän käyttökokemuksesta, sillä sen avulla pyritään mallintamaan käyttäjien interaktiota järjestelmän kanssa. (Stickdorn & Schneider, 2011; Lawrence et al. 2018)

Järjestelmäkartta puolestaan on sateenvarjotermi erilaisille visualisoinneille järjestelmästä. Näitä visualisointeja ovat sidosryhmäkartat, ekosysteemikartat ja arvoverkon kartat. Kaikki mallit voidaan laatia monesta eri näkökulmasta kuten asiakkaan, kilpailijoiden, käyttäjien tai muiden sidosryhmien edustajien näkökulmasta. Järjestelmäkartat sisältävät suhteen muihin palvelumuotoilun metodeihin kuten persooniin ja polkujen mallintamiseen. (Stickdorn et al. 2018) Persoonat voidaan integroida sidosryhminä järjestelmäkarttoihin. Tämä nostaa usein esille erilaisia konflikteja eri asiakasryhmien välillä ja niiden avulla voidaan esittää lyhyesti millaisia sisäiset ja ulkoiset sidosryhmät ovat. (Stickdorn et al. 2018; Stickdorn & Schneider, 2011, ss. 150-151)

Korttien lajittelu ja viisi parasta

Korttien lajittelun tarkoituksena on ryhmitellä käyttäjien vaatimuksia ja niitä tekijöitä, jotka vaikuttavat eniten käyttäjiin suunniteltaessa uutta järjestelmää. Vaatimusmäärittelijän tehtävänä on laatia erilaisia kortteja, jotka sisältävät sanan tai yksittäisen kuvan järjestelmästä. Kortit asetellaan pöydälle ja käyttäjien tulee ryhmitellä kortit tavalla, joka hänelle on selkein ja loogisin. Ryhmittelyn jälkeen käyttäjän tulee kertoa vaatimusmäärittelijälle, miksi hän ryhmitteli kortit juuri tällä tavalla ja millainen logiikka kategorisointiin on. Korttien lajittelu mahdollistaa uudelle järjestelmälle kehitettävän informaatioarkkitehtuurin olevan käyttäjille intuitiivinen ja helposti käytettävä. (Zowghi & Coulin, 2005; Carrizon et al. 2008, s. 110-112) Korttien lajittelu on onnistunut palvelumuotoilumetodi vaatimusmäärittelyyn, jos kortit sisältävät mahdollisimman tarkasti kaikki järjestelmän entiteetit ja vaatimusmäärittelijä ymmärtää ongelmakontekstin hyvin (Carrizon et al. 2008, s. 111).

Viisi parasta (*eng. top five*) metodi toimii korttien lajittelun tapaan kategorisoinnin ja priorisoinnin työkaluna. Menetelmän tarkoituksena on yksinkertaisesti kysyä välillä mitkä ideat tai teemat ovat viisi parasta. Kysymykseen vastaaminen ei ainoastaan auta laatimaan strategiaa järjestelmän toteutuksesta vaan auttaa myös paljastamaan teemoja, rajaamaan ideoita sekä tunnistamaan suunnittelumahdollisuuksia, joita pidetään tärkeimpinä. (Zowghi & Coulin, 2005)

5. KONSTRUKTIO TEORIASTA

Työn kriittinen kirjallisuuskatsaus toteutettiin kahdesta eri näkökulmasta, joita olivat toiminnallinen vaatimusmäärittely ja palvelumuotoilu sekä sitä kautta johdetut palvelumuotoilumetodit. Tässä luvussa on tarkoituksena keskittyä tutkimuksen kirjallisuuskatsauksen perusteella esille nousseihin lähestymistapoihin. Kirjallisuuskatsauksen lähestymistapojen perusteella ensin muodostetaan kappaleessa 5.1 toiminnallisen vaatimusmäärittelyn prosessimalli ja kappaleessa 5.2 tätä prosessimallia rikastetaan erilaisten palvelumuotoilumetodien avulla.

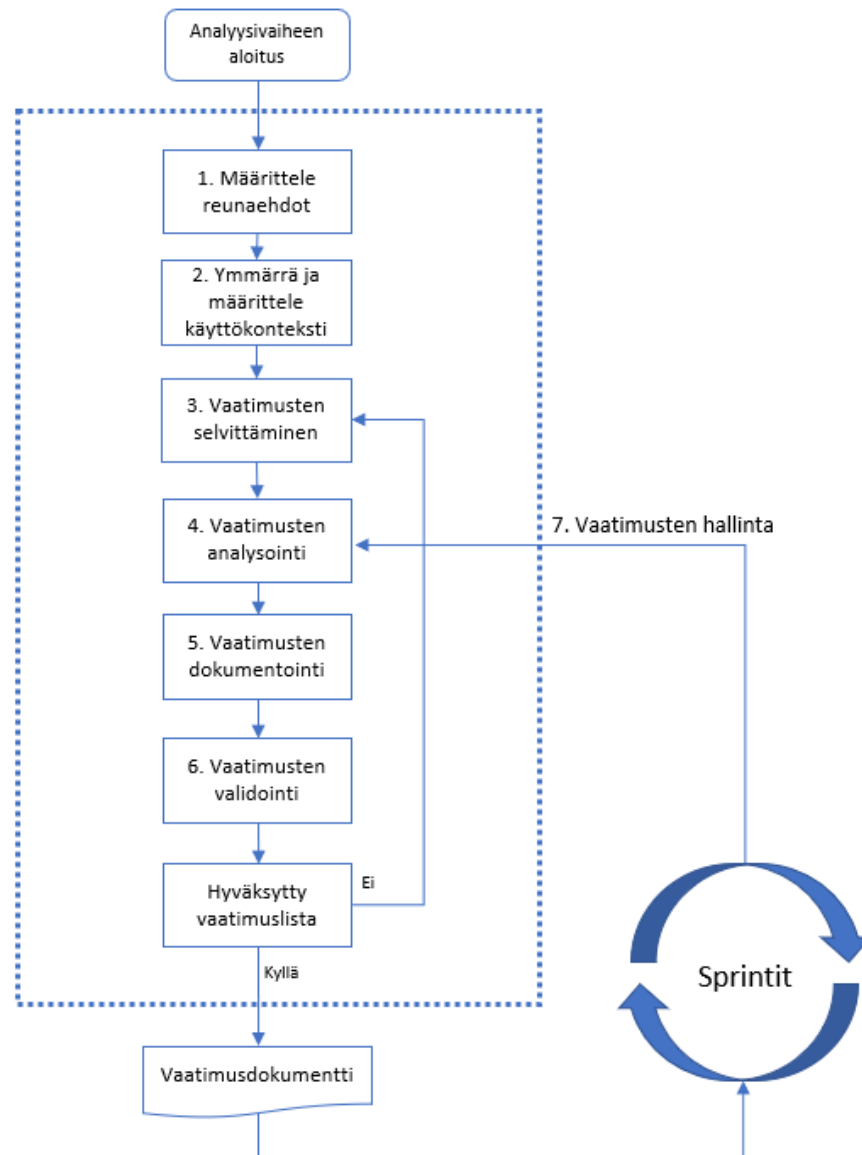
5.1 Toiminnallisen vaatimusmäärittelyn prosessimalli

Työn luvussa 3 on aiemmin esitelty erilaisia menetelmiä toiminnallisten vaatimusten määrittämiseen ja siihen, miten vaatimuksia tulisi mallintaa. Kirjallisuuskatsaus pyrki löytämään myös ihmiskeskeisiä tekijöitä, joiden avulla vaatimusmäärittelyn prosessia voitaisiin kehittää. Taulukkoon 13 on koottu teorian keskeisin sisältö.

Taulukko 13. Kirjallisuuskatsauksen kautta esille nousseet lähestymistavat

Lähestymistapa	Mitä sisältää	Tärkeimmät tekijät
<i>Vaatimusmäärittelyprosessi</i>	Prosessimalli	Iteratiivinen prosessimalli sisältäen onnistumisen kriteerit.
<i>Ketterä vaatimusmäärittely</i>	Periaatteita ja menetelmiä implementoida ketteriin projekteihin ja vaatimusmäärittelyprosessiin	Yhdistää ketterän projektimenetelmän, vaatimusmäärittelyn sekä ihmislähtöisen suunnittelun.
<i>Käyttäjätarinat</i>	Vaatimusten mallintamisen menetelmä	Jaottelu: teema, epiikat ja käyttäjätarinat. Tulee olla itsenäinen, neuvoteltavissa, hyödyllinen, estimoitava, pieni ja testattava.
<i>Ihmiskeskeinen suunnitteluprosessi</i>	Prosessimalli	Iteratiivinen prosessimalli sisältäen ihmiskeskeisen suunnitteluprosessin yleiset periaatteet.

Taulukossa 13 esitettyjen teorioiden pohjalta voidaan johtaa toiminnallisen vaatimusmäärittelyn prosessimalli. Prosessimalli pohjautuu kirjallisuuden avulla esille nousseen vaatimusmäärittelyprosessiin. Vaatimusmäärittelyprosessiin on lisätty ketterän vaatimusmäärittelyn piirteitä sekä ominaisuuksia ihmiskeskeisestä suunnitteluprosessista. Kuvassa 8 on esitetty muodostettu prosessimalli.



Kuva 8. Ihmiskeskeisen toiminnallisen vaatimusmäärittelyn prosessimalli

Prosessimalli lähtee liikkeelle reunaehtojen tunnistamisesta ja etenee vaihe vaiheelta kohti hyväksyttyä vaatimusdokumenttia. Jokaiselle prosessin vaiheelle on tunnistettavissa erilaisia tekijöitä, joiden avulla saavutetaan vaiheille asetetut tavoitteet. Taulukkoon 14 on kuvattu prosessin vaiheittain tapahtuvat tehtävät ja tunnistettavat asiat.

Vaatimusmäärittelyprosessi vaiheessa 1 määrittellään projektin reunaehdot. Tässä vaiheessa prosessia saavutetaan selkeä ymmärrys siitä, millaista projektia ollaan lähtemässä toteuttamaan ja mitä projektilta tavoitellaan. Lopputuloksena reunaehtojen määrittelyn jälkeen saadaan kaavio projektin toteuttamiskelpoisuudesta ja visiosta. (*eng. Viability and Vision Project Document*) Kun projektin reunaehdot ovat tunnistettu siirrytään prosessissa vaiheeseen 2, jossa tunnistetaan ja määritetään käyttökonteksti. Käyttökontekstin määrittäminen lähtee liikkeelle käyttäjien ja sidosryhmien tunnistamisella ja heidän ominaisuuksien analysoinnilla. Tässä vaiheessa selvitetään millaisia ominaisuuksia, tavoitteita ja tarpeita käyttäjillä ja sidosryhmien edustajilla on. Samalla pyritään kokonaisvaltaisesti ymmärtämään järjestelmän ympäristöä. Lopputuloksena tässä prosessin vaiheessa saadaan käyttökontekstikuvaus.

Käyttökontekstin kuvaamisen jälkeen siirrytään prosessin vaiheeseen 3, jonka tarkoituksena on tunnistaa vaatimukset. Tässä vaiheessa selvitetään mitä käyttäjän tulisi saavuttaa järjestelmän käytöllä. Vaatimusten selvittämiseen on tunnistettavissa erilaisia näkökulmia, joiden avulla vaatimuksia pysytään selvittämään. Tärkeimmät näkökulmat on listattu taulukossa 14.

Kun vaatimukset ovat tunnistettu tulee niitä analysoida. Vaatimusmäärittelyprosessin vaiheessa 4 vaatimukset luokitella, priorisoidaan ja niiden toteutettavuus analysoidaan. Kun vaatimuksia kerätään, voidaan löytää paljon mahdollisia poikkeavia tai ristiriitaisia tarpeita sidosryhmien välillä. Tämän prosessivaiheen tarkoituksena on tarkastella vaatimuksia kokonaisvaltaisesti niin, että jokaisen sidosryhmän vaatimukset otetaan huomioon ja ristiriitatilanteet selvitetään. Vaatimukset tulee luokitella samankaltaisuuksien kanssa yhteen ja vaatimukset priorisoidaan vaatimuslistalla. Priorisoinnin tarkoituksena on listata vaatimukset siten, että kaikista tärkeimmät ja oleelliset vaatimukset valitaan ja päätetään missä järjestyksessä ne toteutetaan. Projektien budjetit ja aikataulut ovat rajalliset ja siksi tässä vaiheessa tulee miettiä mitä asioita koetaan hyödyllisiksi ja mitkä toiminnot toteutetaan ensin. Kaikki toiminnallisuudet, joita ei keretä toteuttamaan projektin puitteissa siirretään jatkokehitysvaiheeseen.

Kun vaatimukset on analysoitu ja priorisoitu, saadaan lopputuloksena priorisoitu vaatimuslista. Tässä vaiheessa vaatimukset dokumentoidaan siten, että dokumentti on mahdollisimman selkeä, johdonmukainen sekä toteutettavissa oleva. Vaatimusdokumentti toimii kommunikaation lähteenä sidosryhmien ja kehitystiimin välillä ja siksi on tärkeää, että dokumentti on laadittu kielellä, jota kaikki ymmärtävät ja sitä on selkeää seurata.

Dokumentointivaiheessa keskitytään pelkästään mallintamaan käyttäjätarinoita selkeästi ja johdonmukaisesti. Koska kyseessä on ketterä ohjelmistoprojekti vaatimukset muuttuvat jokaisella sprintillä. Dokumentit muuttuvat nopeasti tarpeettomiksi ja siksi dokumenttia tulee päivittää koko ketterän ohjelmistoprojektin toteutuksen ajan. Lopputuloksena tässä prosessinvaiheessa saadaan vaatimusdokumentti. Vaatimusdokumentti tulee kuitenkin jokaisessa vaatimusmäärittelyprosessissa validoida ennen kuin vaatimuksia voidaan lähteä toteuttamaan. Vaatimusten validoinnissa tarkistetaan, että vaatimukset täyttävät kaikki laadulliset kriteerit. Kun vaatimukset on käyty läpi, voidaan vaatimuslista hyväksyä. Tässä prosessin vaiheessa kaikkien relevanttien sidosryhmien tulee olla jatkuvasti läsnä ja heidän tulee hyväksyä lopulliset käyttäjävaatimukset ennen kehitysprojektin alkua. Vaatimusmäärittelyprosessi loppuu vaiheeseen 6 jossa vaatimukset validoidaan ja hyväksytään. Tämän jälkeen voidaan siirtyä analyysivaiheesta projektin toteutukseen.

Kun prosessimallia tarkastellaan lähemmin, on hyvä kiinnittää huomiota siihen, että ketterissä projekteissa koostetaan aluksi käyttäjätarinoista tuotteen kehitysjohto, jota käsitellään myöhemmin tarkemmin sprinttien yhteydessä. Tarkemmat määritelmät toteutuksesta ja toteutettavista ominaisuuksista tulee käsitellä vasta sprintin suunnittelupalaverissa ja tässä vaiheessa esille nousseet muutokset tulee päivittää tuotteen kehitysjohtolle. Ketterälle projektille on hyvin tyypillistä, että vaatimukset muuttuvat vielä hyvin myöhäisessäkin vaiheessa kehitysprosessia ja siksi kaikki muutokset tulee kirjata kehitysjohtolle heti kun niitä havaitaan. (Beck et al. 2001; Schwaber & Beedle, 2001; Schwaber & Sutherland, 2017)

Koska kyseessä on ketterä ohjelmistoprojekti, on tärkeää muistaa, että vaatimusten hallintaa tulee tapahtua jatkuvasti kehitysprojektin ajan ja siksi se tulee ottaa huomioon prosessimallissa. Prosessin vaihe 7 keskittyy vaatimusten hallintaan kokonaisvaltaisesti. Ketterät ohjelmistoprojektit kannustavat siihen, että muuttuviin tilanteisiin reagoidaan ja asioita kehitetään jatkuvasti. Esimerkiksi jos kehitystiimi huomaa toteutusvaiheessa, että jokin vaatimusdokumentilla oleva vaatimus on liian kompleksinen, tulee se jakaa pienempiin osiin. Tässä tilanteessa kompleksinen vaatimus palautetaan sprintiltä takaisin vaatimusmäärittelyprosessille ja vaatimusta lähdetään analysoimaan uudelleen. Myös muut pienemmät muutokset vaatimuksissa päivitetään kehitysjohtolle, jotta projektin lopussa vaatimusdokumentti on kattava dokumentti projektin kaikista vaatimuksista.

Taulukko 14. Ihmiskeskeisen toiminnallisen vaatimusmäärittelyn tunnistettavat tekijät

Prosessin vaihe	Tunnistettavat asiat	Lopputulos
1. <i>Reunaehtojen määrittäminen</i>	<ul style="list-style-type: none"> • liiketoimintatavoitteet • Dokumentoinnin tarkkuus ja tavoitteet • Järjestelmän luonne • Järjestelmän kehitysympäristö 	Projektin toteut- tamiskelpoisuus- ja visiokaavio
2. <i>Käyttökontekstin määrittäminen</i>	<ul style="list-style-type: none"> • Käyttäjät ja sidosryhmät • Käyttäjien ja käyttäjäryhmien ominaisuudet • Käyttäjien tavoitteet ja tehtävät • Järjestelmän ympäristö 	Käyttökonteksti- kuvaus
3. <i>Vaatimusten selvittäminen</i>	<ul style="list-style-type: none"> • Suunniteltu käyttökonteksti • Vaatimukset, jotka on johdettu käyttökontekstista ja käyttäjätarpeista • Vaatimukset, jotka on johdettu käyttöliittymästandardeista, -ohjeistuksista ja -tietämyksestä • Vaatimukset ja tavoitteet, jotka liittyvät käytettävyyteen • Mitattavat suoritus- ja kyvykkyysskriteerit määritellyssä käyttökontekstissa • Järjestelmävaatimukset, jotka pohjautuvat organisaation tarpeisiin ja joilla nähdään olevan vaikutusta järjestelmän käyttäjiin 	Käyttäjätarvoin kuvatut vaati- mukset
4. <i>Vaatimusten analysointi</i>	<ul style="list-style-type: none"> • Vaatimusten luokittelu • Vaatimusten priorisointi • Vaatimusten toteutettavuuden analysointi 	Priorisoitu vaati- muslista
5. <i>Vaatimusten dokumentointi</i>	Dokumentin tulee olla: <ul style="list-style-type: none"> • Selkeä • Johdonmukainen • Toteutettavissa oleva 	Vaatimusdoku- mentti
6. <i>Vaatimusten validointi</i>	Vaatimusten laadun tarkistus: <ul style="list-style-type: none"> • Itsenäinen – yksi lause • Neuvoteltavissa oleva • Hyödyllinen • Estimoitava • Pieni • Testattava 	Validoitu ja hy- väksytty vaati- musdokumentti
7. <i>Vaatimusten hallinta</i>	Jatkuva iterointi: <ul style="list-style-type: none"> • Tiheät julkaisut – mahdollistaa palautteen keruun käyttäjältä • Vaatimusten evoluutio – muutokset vaatimuksissa päivitetään kehitysjonolle 	Jatkuva kehiti- mistä

- Vaatimusten jakaminen osiin – jos vaatimukset liian kompleksisia
- Vaatimusten jäljitettävyyden – vaatimusten tulee olla toteutettavissa myös tulevaisuudessa vaatimusdokumentin perusteella

Luvussa kolme nostettiin esille erilaisia ketterän vaatimusmäärittelyn menetelmiä, joita käytetään vaatimusten esille saannissa. Ketterän vaatimusmäärittelyn erityispiirteet tulee pitää mielessä, jotta vaatimusmäärittelystä saadaan tehokasta. Koko vaatimusmäärittelyprosessin ajan on tärkeää osallistaa käyttäjiä ja sidosryhmien edustajia aktiivisesti. Tärkeät ja relevantit sidosryhmät tulee tunnistaa sekä jokaisessa prosessin vaiheessa tulee kunnioittaa heidän vaatimuksiansa. Erilaisten menetelmien ja työkalujen käyttö mahdollistaa sen, että sidosryhmiltä saadaan kerättyä mahdollisimman paljon tietoa, jota käytetään koko määrittelyprosessin ajan.

5.2 Prosessimallin kehittäminen palvelumuotoilumetodeilla

Työn toinen teoriaosuus keskittyi luvussa 4 selvittämään mitä palvelumuotoilulla tarkoitetaan ja millaisia metodeja palvelumuotoilulle on tunnistettavissa. Kappaleessa 4.2 esiteltiin erilaisia palvelumuotoilumetodeja, joilla voidaan tukea toiminnallista vaatimusmäärittelyprosessia. Taulukkoon 15 on koostettu tärkeimmät palvelumuotoilumetodit vaatimusmäärittelyprosessia ajatellen sekä selvennetty eri metodien käyttökontekstia.

Taulukko 15. *Palvelumuotoilumetodit vaatimusmäärittelyprosessissa*

Prosessivaihe	Metodi	Käyttökonteksti
1. <i>Reunaehtojen määrittäminen</i>	Haastattelut	Nykyisen ja tulevan tilan selvitys
	Kohderyhmä	Valitun aiheen syvällisempi tarkastelu (haastattelumetodi tarvittaessa)
	5 x Miksi	Haastattelutekniikka (tarvittaessa)
	Aivorihi	Kompleksisen ongelmatilanteen ymmärtäminen
2. <i>Käyttökontekstin määrittäminen</i>	Haastattelut	Määriteltävien asioiden selvittäminen
	Kohderyhmä	Valitun aiheen syvällisempi tarkastelu (haastattelumetodi tarvittaessa)
	5 x Miksi	Haastattelutekniikka (tarvittaessa)
	Yhteistyöpajat	Yhteisen tekemisen mahdollistaminen
	Sidosryhmäkartat	Sidosryhmien tunnistaminen ja priorisointi
	Persoonat	Todellisten tarpeiden ja tavoitteiden tunnistaminen
	Asiakkaan polku	Järjestelmän käyttökokemuksen mallintaminen

3. <i>Vaatimusten selvittäminen</i>	Yhteistyöpajat	-
	Käyttäjätarinat	Inhimillistetaan käyttäjien vaatimuksia, jotta niihin voidaan samaistua. Auttaa ymmärtämään käyttäjien tarpeita.
	Skenaariot	Käyttäjätarpeiden ymmärtäminen
	Haastattelut	Vaatimusten selvittämisen tukena
4. <i>Vaatimusten analysointi</i>	Korttien lajittelu	Vaatimusten ryhmittely
	Viisi parasta	Vaatimusten ryhmittely ja priorisointi
5. <i>Vaatimusten dokumentointi</i>	-	-
6. <i>Vaatimusten validointi</i>	-	-
7. <i>Vaatimusten hallinta</i>	-	-

Palvelumuotoilumetodeilla voidaan rikastaa aikaisemmin esiteltyä vaatimusmäärittelyn prosessimallia. Menetelmistä osa sopii hyvin moneen eri prosessin vaiheeseen ja menetelmiä voidaan käyttää monella eri tapaa ja niitä voidaan lähestyä monesta eri näkökulmasta. Erityisen hyvin palvelumuotoilun metodit sopivat vaatimusmäärittelyprosessin alkuvaiheisiin (vaiheet 1-4), sillä metodien avulla pystytään selvittämään ja määrittämään prosessin alkuvaiheen asioita hyvin ketterästi kuten reunaehdoja ja käyttökonteksteja. Palvelumuotoilumetodit ovat usein osallistavia, joiden tarkoituksena on saada mahdollisimman paljon näkemyksiä juuri eri vaiheiden asiantuntijoilta ja näkemyksiä rikastetaan yhteisen tekemisen kautta.

Vaatimusmäärittelyprosessin vaiheessa 1 keskitytään reunaehtojen määrittelyyn. Tähän prosessivaiheeseen sopivat erityisesti menetelminä haastattelut ja aivoriihi. Jos käynnistettävä projekti tunnistetaan heti haastavaksi, voidaan aivoriihen avulla ymmärtää kompleksisia ongelmatilanteita paremmin ja tässä vaiheessa projektia voidaan myös rajata tarkemmin sitä, mitä myöhemmin on mahdollista toteuttaa ja mitä voidaan nähdä jätettäväksi jatkokehitykseen. Haastattelut puolestaan toimivat prosessin ensimmäisessä vaiheessa aina hyvänä metodina. Haastatteluiden avulla laajennetaan tietämystä valitusta aiheesta. Tarkoituksena on toteuttaa haastattelut puoliavoimina jopa hyvin keskusteluvina tilaisuuksina ja haastatteluiden aikana voidaan käyttää tarvittaessa hyväksi erilaisia haastattelumenetelmiä kuten kohderyhmää tai 5 x Miksi -tekniikkaa. Erityisesti 5 x Miksi -tekniikka pureutuu ongelmien ja tekijöiden juurisyihin ja haastattelijan on hyvä tiedostaa metodin olemassaolo. Esittämällä miksi-kysymyksiä haastattelun aikana pystytään pureutumaan aiheeseen paljon syvällisemmin.

Vaatimusmäärittelyprosessin vaiheessa 2 voidaan käyttää palvelumuotoilumetodeista haastatteluita ja yhteistyöpajoja. Yhteistyöpajojen aikana tuotetaan tilanteeseen sopivia

dokumentteja ja hyödynnetään erilaisia palvelumuotoilumetodeja. Tässä prosessivaiheessa hyödynnetään sidosryhmäkartoja tärkeimpien sidosryhmien tunnistamiseksi ja priorisoimiseksi. Sidosryhmäkartoista voidaan johtaa erilaisia persoonakuvauksia, joiden avulla pystytään tunnistamaan käyttäjän todellisia tarpeita ja tavoitteita. Kun sidosryhmät ja todelliset tarpeet on tunnistettu, voidaan asiakkaan polkua mallintaa ja keskittyä erityisesti järjestelmän käyttökokemuksen mallintamiseen.

Prosessin vaiheessa 3 selvitetään vaatimuksia haastatteluiden, käyttäjätarinoiden, skenaarioiden ja yhteistyöpajojen avulla. Haastatteluiden aikana laajennetaan tietämystä aiheesta ja pyritään löytämään mahdollisimman paljon erilaisia vaatimuksia. Vaatimuksia mallinnetaan skenaarioiden avulla ja niistä muodostetaan selkeitä yksinkertaisia käyttäjätarinoita, joihin voidaan paneutua toteutusvaiheessa. Tässä vaiheessa on erityisen tärkeää ylläpitää jatkuvaa kommunikointia aikaisemmin tunnistettujen sidosryhmien edustajien ja projektitiimin välillä ja siksi yhteistyöpajat toimivat hyvänä metodina. Niiden aikana tuotetaan yhdessä tärkeiden sidosryhmien kanssa juuri spesifejä skenaarioita sekä käyttäjätarinoita. Tämän avulla mahdollistetaan se, että jokaisten sidosryhmien edustajien mielipiteet ja vaatimukset otetaan huomioon ja, että tässä prosessivaiheessa pystytään tunnistamaan kaikki vaatimukset riippuen niiden toteutettavuudesta tai prioriteetista.

Vaatimusmäärittelyprosessin neljännessä vaiheessa analysoidaan vaatimuksia. Tässä vaiheessa palvelumuotoilumetodeista korttien lajittelu sekä viisi parasta auttavat vaatimusten ryhmittelyssä ja priorisoinnissa. Metodien tarkoituksena on lajitella vaatimuksia loogisiin ryhmiin ja auttaa priorisoimaan vaatimuksia vaatimuslistalla. Prosessivaiheen lopputuloksena saadaan priorisoitu vaatimuslista ja siksi nämä palvelumuotoilumetodit tukevat prosessivaihetta ja auttavat projektitiimiä sekä sidosryhmiä kommunikoimaan yhtenäisesti.

Kun palvelumuotoilumetodit yhdistetään vaatimusmäärittelyn prosessimalliin, voidaan mallia rikastaa erilaisilla pienemmillä tavoitteilla, mukaan otettavilla henkilöillä ja tukidokumenteilla, joita muodostetaan eri menetelmien käytön yhteydessä. Liitteessä 1 on esitetty vaatimusmäärittelyn rikastettu prosessimalli. Työn seuraavassa vaiheessa keskitytään paremmin vaatimusmäärittelyn rikastettuun prosessimalliin ja sitä testataan empirisen tutkimuksen avulla. Tarkoituksena on testata mallia sekä kehittää sitä vielä eteenpäin toteutetun analyysin kautta.

6. EMPIIRISEN TUTKIMUKSEN ANALYYSI JA TULOKSET

Tässä luvussa käsitellään empiirisen tutkimuksen toteutuksen kautta esille nousseita asioita sekä tutkimuksen tuloksia. Empirian tarkoituksena oli testata teorian pohjalta muodostettua mallia sekä haastatteluiden aikana esille nousseiden asioiden avulla mallia kehitettiin vielä eteenpäin.

6.1 Haastattelututkimuksen toteutus

Haastattelut toteutettiin kohdeyrityksen omille työntekijöille, joten heitä lähestyttiin sähköpostilla, jossa esiteltiin tutkimuksen aihe ja haastatteluiden tarkoitus. Kaikki kysytyt haastateltavat osallistuivat haastatteluihin mielellään ja näkivät paljon arvoa haastatteluissa. Tutkimuksen haastatteluita toteutettiin yhteensä 6 kappaletta. Haastatteluiden kohderyhmä rajautui kolmeen näkökulmaan: projekti, tekninen ja konsultti. Osa-alueista haastateltiin kahta näkökulman edustajaa. Haastattelut toteutettiin noin kahden viikon mittaisella ajanjaksolla. Taulukossa 16 on esitetty haastateltavien otanta ja jokaisen haastateltavan kokemustasot.

Taulukko 16. Haastateltavien otanta

Haastateltava	Näkökulma	Kokemustaso
<i>H 1</i>	Projekti	1,5 vuoden kokemus ensin ratkaisukonsulttina ja sitten pikkuhiljaa projektipäällikkönä.
<i>H 2</i>	Projekti	Kolmen vuoden kokemus projektipäällikön tehtävistä ja analyysivaiheen toteutuksesta.
<i>H 3</i>	Tekninen	Neljän vuoden kokemus teknisen arkkitehdin tehtävistä.
<i>H 4</i>	Tekninen	3,5 vuotta ratkaisukonsulttina, mutta toiminut myös projekteissa teknisenä arkkitehtina.
<i>H 5</i>	Konsultti	9 vuoden kokemus ratkaisukonsulttina ja tuoteasiantuntijana.
<i>H 6</i>	Konsultti	11 vuoden kokemus ratkaisukonsultin tehtävistä. Toiminut tuoteasiantuntijana koko ajan.

Haastatteluiden kysymyksiä ei lähetetty etukäteen haastateltaville vaan aikaisemmin lähetetyssä sähköpostissa kerrottiin, mihin aiheeseen haastattelut keskittyvät ja mitä haastatteluiden avulla halutaan selvittää. Tämän pohjustusviestin takia haastattelukysymyksiä ei lähetetty haastateltaville etukäteen. Haastatteluista suurin osa toteutettiin kasvotusten ja ne nauhoitettiin. Kiireisten aikataulujen takia muutama haastattelu pidettiin Skypellä ja ne taltioitiin myös nauhoittamalla haastattelu. Kaikki haastattelut toteutettiin yksilöhaastatteluina.

Haastatteluiden aluksi käytiin läpi haastatteluiden tarkoitus sekä kysyttiin hyväksyntä haastattelun nauhoittamiselle. Tämän lisäksi painotettiin erityisesti sitä, että haastatteluiden aikana haastateltavan omat näkemykset ja kokemukset ovat tärkeitä ja niistä saa puhua vapaasti. On myös hyvä, jos haastateltava kyseenalaistaa kirjallisuuden perusteella muodostettua mallia. Tämä mahdollisti sen, että näkemykset olivat hyvin laajoja ja mallin testaaminen sitä kautta luotettavampaa.

Haastattelun kysymykset käytiin läpi haastattelurungon mukaisesti (liite 2). Koska haastattelu toteutettiin puolistrukturoituna teemahaastatteluna, pyydettiin vastauksiin tarkennuksia ja joissain tilanteissa tartuttiin haastateltavan näkökulmiin, jos ne vaikuttivat tutkimuksen kannalta oleellisilta. Haastateltavia pyydettiin esimerkiksi tarkentamaan jotain vastausta, jos haastateltava ei täysin ymmärtänyt mitä haastateltava tarkoitti tai jos haastateltavasta tuntui siltä, että aihe on tutkimuksen kannalta tärkeä. Pääsääntöisesti haastatteluiden aikana saatiin kattavia vastauksia vaatimusmäärittelyprosessin nykytilanteesta ja siitä kuinka uusi malli voisi toimia yrityksen toteuttamissa analyysivaiheissa. Teemahaastattelu mahdollisti sen, että haastateltavat kertoivat hyvin avoimesti omista tuntemuksista ja kokemuksista haastattelun aikana ja tilaisuuden aikana saatiin kerättyä paljon laadukasta materiaalia tutkimuksen aiheesta.

Kaikki haastattelut onnistuivat hyvin ja kaikki keskustelut haastateltavien kanssa olivat rikkaita. Haastatteluihin varattiin aikaa tunti ja siinä ajassa ehdittiin käymään kaikki haastattelun asiat läpi. Tunnin mittainen haastattelu onnistui hyvin, sillä haastattelija oli suunnitellut tilaisuuden toteutuksen etukäteen, ja jokainen haastateltava oli tutustunut etukäteen lähetettyyn sähköpostiin, joka pohjusti haastatteluiden aiheita. Tämä mahdollisti sen, että haastatteluiden aikaa ei tarvinnut pidentää, vaan kaikki tarvittavat asiat käytiin läpi sovitun haastattelutilaisuuden aikana.

Haastatteluiden lisäksi tutkimuksessa käytettiin yrityksen sisäisiä dokumentteja. Osa materiaaleista käsitteli ketterän ohjelmistoprojektin toteutustapaa yrityksessä ja sen vertailua muihin toteutustapoihin. Dokumentin avulla saatiin lähinnä tietoa siitä, miten ketterä projekti toteutetaan kohdeyrityksessä ja miten siitä viestitään asiakkaalle. Osa materiaaleista on myös käytössä projektien myyntivaiheessa. Tämän lisäksi työssä käytettiin analyysivaiheen dokumentteja, jotka auttoivat ymmärtämään paremmin kohdeyrityksessä toteutettavia projektien analyysivaiheita sekä dokumentteja, joihin tukeudutaan, kun analyysivaihetta toteutetaan.

6.2 Aineiston analysointi

Kun oli pidetty haastattelut, joiden avulla laajennettiin tietämystä tutkimuksen aiheesta, pystyttiin siirtymään kerätyn aineiston analysointiin. Haastatteluaineistoa analysoitiin temaattisella analyysillä sekä summaus ja kategorisointi -lähestymistavalla (Braun & Clarke, 2006; Saunders et al. 2009, ss. 491-492). Temaattisen analyysin tarkoituksena on tunnistaa, analysoida ja raportoida säännönmukaisuuksia kerätystä datasta. Temaattisen analyysin avulla huomioidaan tilanteiden riippuvuutta ja tulkinnallisuutta kontekstissa. (Braun & Clarke, 2006; Vaismoradi et al. 2013) Tutkimuksen konstrukttiivinen tutkimusote ohjaa myös temaattisen analyysin toteutusta. Analyysissa etsitään juurisyitä toiminnalle eri tilanteissa sekä analysoidaan toiminnan rakenteellisia piirteitä. (Braun & Clarke, 2006)

Datan summaus ja kategorisointi toimii yleisenä laadullisena analyysimetodina, joka on sovellettavissa laadulliselle datalle. Metodi on monialainen ja sille ei ole tunnistettavissa selkeää analyysiprosessia. (Saunders et al. 2009, ss. 490-491) Metodi perustuu datan käsittelyyn. Tarkoituksena on kerätä tärkeimmät asiat yhteen ja summataan ne eri kategorioiden ja pääteemojen mukaisesti. Eri kategoriat ja teemat liittyvät jollain tavalla dataan ja kuvailevat sitä. Summaaminen kerää yhteen haastatteluiden avainkohdat ja pitkät toteamukset tiivistetään sisältämään oleelliset asiat vain muutamaaan sanaan. (Kvale, 1996; Saunders et al. 2009, s. 491) Summaaminen voi olla haastavakin vaihe aineiston analysoinnissa, sillä haastatteluiden tuloksia tulee käydä huolellisesti läpi, jotta löydetään ne oleelliset asiat ja tunnistetaan haastatteluiden aikana esille nousseet pääteemat. Onnistunut summaaminen kuitenkin auttaa tunnistamaan haastatteluiden aikana esille nousseita laajempia kokonaisuuksia. Samalla vahvistuu näkemys siitä, kuinka haastateltavat voivat nähdä asioita eri tavalla riippumatta lähtökohdista.

Datan kategorisointi tapahtuu kahdessa osassa: kategorioiden luominen ja kategorioiden liittäminen merkitykselliseen dataan. Kun kategorioita luodaan, pyritään niistä tekemään erilaisia nimikkeitä tai koodeja, joiden avulla datan jaottelu ja luokittelu on helpompaa. Erilaiset kategoriat voidaan luoda kirjallisuudesta tunnistetun viitekehyksen kautta tai tunnistamalla niitä haastatteluiden aineistojen perusteella. (Saunders et al. 2009, ss. 491-493)

Temaattinen analysointi etenee tutkimuksessa hyvin suoraviivaisesti, sillä analyysille on tunnistettavissa selkeästi kuusi erilaista vaihetta (Braun & Clarke, 2006; Novell et al. 2017; Vaismoradi et al. 2013). Näitä vaihteita ovat:

1. Aineiston litterointi
2. Koodiston luominen
3. Teemoittain koodatun aineiston kokoaminen
4. Aineiston ja teemojen uudelleen arviointi
5. Teemojen nimeäminen ja jalostus
6. Raportin tuottaminen

Haastatteluiden aikana tehtiin haastattelun etenemisen kannalta tärkeimpiä muistiinpanoja, jotka ohjasivat haastattelijaa kysymään johdonmukaisia jatkokysymyksiä. Kaikki haastateltavat suostuivat siihen, että haastattelut nauhoitettiin ja siksi haastatteluiden litterointia oli suoraviivaista tehdä. Haastatteluita ei litteroitu sanatarkasti, ellei vastaus vaikuttanut niin merkittävältä tutkimuksen kannalta ja siinä olisi pystytty kiteyttämään sanoja ja sanojen tarkoitusta sopivalla tavalla. Litteroinnin aikana pyrittiin luomaan mahdollisimman todellinen kuva mahdollisimman tiiviisti kirjoitettuna. Litteroinnissa oli tarkkaa kuitenkin analysoida ja miettiä haastateltavien mahdollisia epäröintejä ja uudelleen aloituksia ja näitä otettiin huomioon osittain litteroinnin aikana.

Kun haastattelumateriaalit oli litteroitu, pystyttiin luomaan koodisto kerätyille materiaaleille. Kategoriat tutkimukselle muodostettiin kirjallisuuden pohjalta, sillä haastattelurunko muodostui kirjallisuuden teemojen ja laaditun prosessimallin pohjalta. Kirjallisuudesta tunnistetut kategoriat olivat prosessimallin vaiheet eli reunaehdot, käyttökonteksti, vaatimusten määrittäminen, analysointi, dokumentointi, validointi ja ylläpito.

Haastatteluaineisto käytiin läpi kahteen kertaan. Kun haastattelut oli litteroitu ja summattu, käytiin jokainen haastattelu läpi sekä jokainen summattu asia luokiteltiin eri kategorioihin. Jos kirjallisuuden kautta tunnistetut kategoriat eivät riittäneet ja haastatteluma-

terიაალები ნოსი ესილე ჟოტაინ თეეოჟა, ჟოტა ეი პუსტყტი სიჟოტტამაან აიკაისემინ ლოტუი-
ჰინ კატეგორიოჰინ, ლოტიინ უსია კატეგორიოტა, ჟილა სუმმატუჟა მატერიაალეჟა პუსტყტიინ ლო-
კიტელემაან.

კუნ კაიკი ჰაასტატელუტ ოლი კაყტი ლაჟი, ტივისტეტი სოჟივან მუოტონ სეკა კატეგორიოიუ ლო-
გისესი, კაყტიინ ჰაასტატელუდინ ტულოკსე კოკონაისუდესაან ლაჟი. ტაჟსა ვაიჰეესა პირი-
ტიინ ტუნისტამაან სუერეჟია თეეოჟა, ჟოტა ოლივ ტუტკიმუკსენ კანალტა ტარეკიჟა. ტუნის-
ტეტუჟენ თეეოჟენ სეკა ვასტაუსტენ ნაკოკულმიენ პოჟალტა პუსტყტიინ მუოდოსტამაან უსი
კეჰიტეტი პროსესიმალი, ჟოკა სისალტაა ტაყდენტავია ნაკოკულმია, ჟოტა ნოსივან ესილე ჰაას-
ტატელუტუტკიმუკსენ აიკანა.

6.3 Tutkimuksen tulokset

ტუტკიმუკსენ ტულოკსე ვოდაან ჟაკაა ჰაასტატელუდინ ჟა პროსესიმალინ ავულა ჰელომინ
კასიტლტავიინ ოსიინ. ნაჟიტა ოსია ოვან ნყკყტილა, ესისელვყს, ტუოტეენ კეჰყსჟონო ჟა ვაატიმუ-
სტენ ჰალინტა. სეურაავისა კაჟჟალეისა კაყდაან ტარეკმინ ლაჟი ტუტკიმუკსენ ტულოკსია ტუნ-
ისტეტუჟენ აიჰეალუეიდენ კაუტა.

6.3.1 Nykytila

ჰაასტატელუდინ აიკანა ვაჰვისტუი ნაკემყს სიჟა, ეტა ვაატიმუსმარეტილე სეკა ანალიყსი-
ვაიჰეე ოვან კოჰდეყრყკსეჟსა კოეტუ ტარეკიკსი ტეკიჟოიკსი ერყყსესი ისოისა პროჟექტეისა.
ანალიყსივაიჰე ლაჰტეე ლიკეელე ტუნისტამალა ასიაკკაან ტარეკიმატ პააჟროსესიტი ჟა ანა-
ლიყსოიმალა სიჟა, მილასია ტოიმინნალისუოკსია ტულეე ტოტეუტაა ჟა მილკ კომპონენტეილა ნამა
პროსესიტი ვოდაან მალინტაა. ანალიყსივაიჰეენ ტარკოტუკსენა ონ ოლუტ ედეტა პააჟროსესიენ
კაუტა პიკკუჰილჟაა ყკსყყსოკოტაისემალე ტასოლე. ონსიმამაისტენ ვაატიმუსმარეტილეიდენ
კანსა ეი ოლუტ ტარკოკოჟა მენეტელმია ტაი მალეჟა, ჟოტა ნოუდატაა. ტიედეტიინ, მიტა ჰალუტაან
საადა ლოჟუტულოკსენა ჟა ნიიდენ პერუსტეელა ანალიყსივაიჰეიტა ტოტეუტეტიინ. ონსიმამაისტენ
ტოტეუტუსტენ ჟალკეენ ჰავაჟიტიინ ასიოტა, ჟოტა ტული პარანტაა ჟა კეჰიჟტაა ეტეენჟაინ. ნყკყსიინ
ანალიყსივაიჰეენ მალი ტოიმი ენემამან მარამუოტოისენა ჟა ოჟაილევანა კუინ ონსიმამაი-
ნენ ვერსიო. სუური ოსა ჰაასტატელტავისა ოლი სიჟა მიელტა, ეტა ანალიყსივაიჰეენ ტოტეუტუს ონ
ელინჰო ისოილე პროჟექტეილე. ანალიყსივაიჰეენ აიკანა კერეტაან პანეუტუმაან ვაატიმუკსიინ
ჟა კოეტიინ, ეტა ასიაკკაან ტეკემილკ ესიტეჟტავილკ ონ სუური როლი ვაიჰეენ ონნისტუმისენ კან-
ნალტა. (H1, H2, H4, H5, H6)

*”ტუნტუი ჟოჟა ჰულლულტა ნყტ მიელტიჟა, ეტა ლაჰდეტაჟსიინ ტოტეუტტამაან ნაინ ისოა პროჟექტია ილ-
მან სისტემაატისტა ანალიყსივაიჰეტა. ანალიყსივაიჰე ანტაა ჰყვან ვალმიუდეტ ტოტეუტტაა
პროჟექტი ონნისტუნეესი.” (H2)*

Tähän asti analyysivaiheisiin on osallistunut projektipäällikkö, tekninen arkkitehti ja ratkaisukonsultti. Jokaisen haastateltavan mielestä analyysivaiheeseen tulee osallistua vähintään edellä mainitut henkilöt. Aikaisempien analyysivaiheiden onnistuminen on ollut pitkälti kiinni projektitiimin osaamisesta ja kokemuksesta. H2, H4 ja H5 nostavat esille, että on tärkeää, että analyysivaiheen projektitiimissä on mukana ratkaisukonsultti, joka tietää paljon asioista ja osaa kertoa näkemyksiä moniin eri tekijöihin. H1 mukaan on onnistumisen kannalta hyvä, että analyysivaiheeseen osallistuu henkilö, joka osaa kertoa järjestelmän kustomoinneista ja asiakkaalle näkyvistä asioista. Tämän lisäksi tulee olla mukana myös tekninen osaaja, joka osaa kertoa miten asioita toteutetaan ja millaisia integraatioita voidaan tehdä. H2 korosti erityisesti sitä, että ratkaisukonsultin tulee olla sellainen henkilö, joka tuntee järjestelmän hyvin ja tietää kaikki mahdollisuudet mitä järjestelmällä voidaan tehdä ilman ohjelmointia. Isojen projektien kannalta teknisen osaajan osallistuminen on välttämätöntä, sillä kyseessä on niin isoja kokonaisuuksia. (H1, H2, H3, H4)

On kuitenkin huomattava, että analyysivaiheen aikana erityisesti isojen projektien kohdalla projektitiimin osaaminen ei aina riitä kattamaan käyttäjien vaatimuksia. H5 ja H6 nostivat esille, että olisi hyvä, jos eri osa-alueiden osaajien resursseja olisi osittain käytössä analyysiprosessin aikana. Isoissa projekteissa muutaman tunnin konsultointi muiden alueiden osaajilta helpottaisi vaikeiden kokonaisuuksien hallintaa.

Nykyinen analyysivaiheen malli koettiin yleisesti onnistuneeksi. Hyvänä asiana pidettiin sitä, että on olemassa standardit dokumenttipohjat, joista voidaan lähteä liikkeelle. Toisaalta analyysivaiheen alussa prosessit saadaan kuvattua hyvin ja jos aikaa on riittänyt, on vaatimukset pystytty kirjoittamaan niin tarkasti, että niitä voidaan alkaa toteuttamaan. Kuitenkin, jos aika on loppunut kesken, on se heti huomattavissa vaatimusten tarkkuustasossa. (H1, H2, H4) Erityisen hyvä on ollut onnistunut sisäinen kommunikaatio sekä se, kuinka näin isoista ja kompleksisista lähtökohdista on pystytty toteuttamaan onnistuneita projekteja, joissa asiakastyytyväisyys on ollut korkeaa. (H3) H5:n mukaan edellytys onnistuneelle analyysivaiheelle on ollut projektitiimin roolit ja niiden tarkka määrittäminen. On tärkeää, että projektipäällikkö osaa ohjata tilanteita ja jarrutella sekä ratkaisukonsulttia, että teknistä arkkitehtiä, jos analyysivaiheessa lähdetään liikaa määrittelyyn.

Ongelmina voidaan puolestaan pitää kiireistä aikataulua, ennakkotehtävien vakinaisuutta, asiakkaiden passiivisuutta ja vaatimusten tarkkuustasoa. Lähes kaikkien haastateltavien mielestä kiireinen aikataulu koetaan suurimmaksi ongelmaksi analyysivaiheen

aikana. Jos kiire on liian kova ei voida olla täysin tyytyväisiä analyysivaiheen lopputulokseen. Kiireinen aikataulu vaikuttaa myös siihen, että asiakkaan puolelta on hankalaa saada kaikkia resursseja mukaan analyysivaiheeseen tarvittavalla tasolla. Vaikka oikeat henkilöt pystyttäisiin tunnistamaan asiakkaan puolelta, ei heitä saada osallistumaan projektiin siinä määrin, mitä projektitiimi toivoisi. Myös haasteellisena asiana pidetään sitä, että analyysivaihe voidaan toteuttaa niin aikaisessa vaiheessa, että itse projektin toteutuksen ja analyysivaiheen välillä on pitkä aikaväli, jolloin kohdataan tilanne, jossa vaatimukset eivät vastaa enää siihen, mitä asiakas haluaa. Ambler (2002), Cohn (2004) ja Collier (2011) ovat tunnistaneeet saman ongelman ja heidän mukaan ketterälle kehitykselle on tärkeää, että vaatimukset mallinnetaan vasta juuri ennen sprinttejä. Mitä lähempänä toteutusta vaatimuksia voidaan mallintaa, sitä helpompi niitä on toteuttaa ja sitä paremmin ne mukautuvat muuttuviin tilanteisiin. Kotaiah ja Khalil (2017) sekä Abdulhalim et al. (2018) korostavat, että ketterän kehityksen tulee mukautua muuttuviin tilanteisiin ja siksi toteutettavaa listaa pidetään dynaamisena sekä tuotevastaavan tulee tehdä päivityksiä listaan aina kun jokin asia muuttuu. Tuoteomistajan rooli kuuluu kohdeyrityksessä projektipäällikölle sekä asiakkaan puolelta nimetylle tuoteomistajalle. Parantaisen (2007, s. 197) mukaan liian pitkälle ennustaminen ketterissä projekteissa ei tue projektin onnistumista ja siksi tulee aina muistaa, että muuttuviin vaatimuksiin tulee reagoida.

Haastateltavien kesken nähtiin kuitenkin analyysivaiheen aikataulu ja resurssit jollain tavalla ristiriitaisina. Erityisesti konsulttinäkökulmasta erot ovat isoja. H6:n mukaan analyysivaiheet on toteutettu tosi hyvin ja niiden aikana on aikaa paneutua vaatimuksiin. Hänelle välittyi tunne, että on aikaa miettiä vaatimuksia eikä projektin etenemisestä synny liikaa paineita. H5 puolestaan korosti haastatteluiden aikana sitä, että analyysivaiheen aikana aikataulu oli liian kiireinen eikä aikaa jäänyt asioiden miettimiseen. Jos analyysivaiheelle on asetettu hyvin tiukka aikataulu, tulisi projektitiimin asettaa selkeä työnjako, jotta vaihe saadaan onnistuneesti toteutettua. Olisi hyvä, jos projektitiimille sekä asiakkaalle jätettäisiin aikaa sisäistää isoja kokonaisuuksia. Jos analyysivaiheen aikana voitaisiin pitää edes pieni hengähdystauko, auttaisi se hankalien kokonaisuuksien käsitteilyssä.

”Analyysivaiheen aikana jouduttiin tekemään paljon rajauksia ja olettamaan, että asiat menevät näin, sillä ei yksinkertaisesti ollut aikaa perehtyä vaatimuksiin enempää. Tiukka aikataulu asetti myös haasteeksi sen, että ei pystytty saamaan muiden alojen osaajilta näkökulmia vaikeisiin vaatimuksiin.” (H5)

Asiakkaan sisäinen kommunikointi on myös osittain puutteellista. Asiakkaan tulisi ennen analyysivaihetta määrittää asioita ja tehdä esitehtäviä, ettei analyysivaiheessa kuluisi aikaa asiakkaan sisäiseen kommunikointiin. Nykyisessä analyysivaiheen mallissa on erilaisia esitehtäviä, joita asiakkaan toivotaan toteuttavan. Näiden merkitys onnistuneen analyysivaiheen kannalta on suuri. H2 mukaan suuri heikkous nähdään siinä, että ei ole vakioituneita ennakkotehtäväpohjia asiakkaille. Jokaisen projektin kohdalla lähdetään tyhjästä tekemään ennakkotehtäviä ja siksi lopputulokset ovat aina hieman erilaisia. Ennakkotehtävät nähdään erityisen hyvänä asiana, mutta niiden roolia voisi korostaa vielä enemmän projekteissa. (H1, H2, H5, H6) Kun asiakas on tehnyt ennakkotehtävät huolellisesti on heillä valmiudet lähteä tekemään analyysivaihetta eikä aikaa kulu liikaa sisäiseen kommunikointiin. Toisaalta H6 muistuttaa, että sisäinen kommunikointi on tärkeää myös projektin aikana, sillä usein kaikki ymmärtävät asioita hieman eri tavalla ja se voi aiheuttaa ongelmia analyysivaiheessa.

Erityisesti H2, H3, H4, H5 ja H6 korostavat, että analyysivaiheessa tulee koko ajan muistaa, että ei mennä vielä määrittämiseen vaan nyt käydään vasta tarpeita läpi. Tässä vaiheessa ei tulisi ottaa kantaa siihen, mitä asioita kannattaisi tehdä vaan enemmänkin keskittyä laajentamaan näkemystä toteutettavasta projektista. Erityisesti Parantainen (2007, s. 204) sekä Girvan ja Paul (2017, ss. 182-185) muistuttavat, että vaatimusten tunnistamisessa on tarkoituksena laajentaa näkemystä käyttäjien tarpeista, eikä siihen, miten palvelun tulisi toimia. Liian tarkka vaatimusten määrittäminen ennen projektin toteutusta ei tue ketterän kehityksen periaatteita.

”Miten voidaan tietää, millä tarkkuustasolla vaatimuksia tulee määrittää. Koska ollaan sillä tasolla, että se riittää työmääräarvioiden tekemiseen, mutta ei mennä liian syvälle määrittelyyn” (H5)

”Ongelmana on se, että tällä hetkellä asioita kirjataan JIRAan vain otsikkotasolla. Näihin tarvitaan projektin myöhemmissä vaiheissa paljon tarkennusta ennen kuin vaatimuksia voidaan lähteä toteuttamaan” (H3)

Girvan ja Paul (2017, ss. 182-185) sekä Leffingwell (2011) tiedostavat myös, että vaatimusten määrittämisen tarkkuustasoa on hankalaa tunnistaa. Liian yksityiskohtainen määrittäminen tekee toteutuksen hankalaksi ja liian tiiviisti ilmaistu vaatimus voi jättää jotain oleellista pois kehitysvaiheessa. Ketterille projekteille ja niiden vaatimusten käyttäjätarinoille voidaan hyvänä ohjeena pitää sitä, että jokaisen käyttäjätarinan tulisi olla

niin tarkka, että se voidaan toteuttaa muutaman viikon aikana eli yhden sprintin kuluessa.

Haastateltavat tiedostivat sen, että projektiryhmä sekä asiakas oppivat projektin aikana paljon ja siksi vaatimuksia ja tarpeita on helpompi käydä läpi projektin edetessä. Luonteeltaan analyysivaihe on kuitenkin sellainen, että sen aikana ei voida kaikkia vaatimuksia tunnistaa ja luokitella. Siksi ison kuvan hahmottaminen analyysivaiheessa on tärkeää. Kiireinen aikataulu heijastuu myös vahvasti tuotteen kehitysjonon (*eng. solution backlogin*) tarkkuuteen ja kehitysjonon osien välillä voidaan havaita suuria tasoeroja. Osa vaatimuksista on määritelty hyvin ja työmääräarviot ovat selkeitä. Vaatimusten seassa on aina kuitenkin niin isoja asioita, joita ei voida käsitellä tarkemmin analyysivaiheessa ja siksi niiden kohdalla usein joudutaan arvaamaan kokemuksen perusteella millainen ratkaisu niihin voidaan toteuttaa. Collierin (2001, ss. 89-91) mukaan tällaisissa tilanteissa on projektitiimin vastuulla se, että isot käyttäjätarinat jaetaan pienempiin osiin viimeistään ennen sprintin alkua, jossa kyseisiä vaatimuksia käsitellään.

H2:n nosti esille kuitenkin kokemuksen merkityksen analyysivaiheiden toteutuksessa. Mitä enemmän analyysivaiheita on toteutettu, sitä järkevämpi kokonaisuus koko prosessista on muodostunut. Myös H4:n on sitä mieltä, että analyysivaihe vaatii tuekseen osaatvat henkilöt, joilla on kokemusta sekä järjestelmän käytöstä, että analyysivaiheen toteutuksesta. Kokemuksella on suuri merkitys onnistumisen kannalta. Myös Parantainen (2007, s. 196-197) vahvistaa tämän näkemyksen, että mitä enemmän kokemusta projektitiimillä on analyysivaiheen toteutuksesta sen tarkempia ja toteutuskelpoisempia sekä käyttäjätarinat, että muut vaatimukset ovat.

6.3.2 Esiselvitys

Esiselvitysvaiheen merkitys analyysivaiheen toteutuksessa koetaan suureksi. Kaikkien haastateltavien mukaan on tärkeää, että lähdetään kehittämään nykyistä analyysivaiheen mallia niin, että käyttäjät tulevat mukaan varhaisemmassa vaiheessa. Uudet järjestelmän ominaisuudet vaativat juuri käyttäjäkeskeisyyttä ja sidosryhmien mallintamista, joten siksi nähdään arvoa siinä, että kehitetään analyysivaiheen mallia tukemaan käyttäjäkeskeisyyttä. Tätä ajatusta tukee ketterän ohjelmistokehityksen muutos suunnitelmapohjaisesta arvopohjaisemmaksi. Ketterien projektien suunnittelussa painotetaan enemmän ihmiskeskeisyyttä ja nykyisin tarkka prosessien määrittäminen on jätetty taka-alalle. (Schön et al. 2016)

Erityisen hyvänä asiana pidetään juuri käyttäjäryhmien selvittämistä ja vaatimusten määrittämistä käyttäjien ja sidosryhmien kautta. Riskinä kuitenkin nähdään se, että asiakkaalle siirtyy paljon vastuuta ja heidän tulee tehdä paljon esiselvitystä ja sitoutua koko analyysivaiheen toteutukseen. Asiakas ei välttämättä ole valmis käyttämään näin paljon aikaa ja resursseja tällaiseen toteutukseen ja siksi koko prosessimalli vaatii tuekseen perusteluita ja materiaaleja, joilla asiakas voidaan saada vakuuttuneeksi analyysivaiheen tärkeydestä.

Erityisesti H2, H4, H5 ja H6 painottivat sitä, että vaatimusmäärittelyn mallia tulisi kehittää vielä eteenpäin siten, että toiminnalliset prosessit tulisi määrittää käyttäjäryhmien kautta. Vaikka tarkoituksena on lähteä liikkeelle käyttäjä- ja sidosryhmistä ei toiminnallisia prosesseja voida kokonaan unohtaa. Tulevaisuudessa on tärkeää, että nämä molemmat näkökulmat otetaan analyysivaiheessa huomioon. Esiselvitysvaihe korostaa ennakotehtäviä ja niiden merkitystä. Jotta kehitetty prosessimalli toimisi haastateltavien mielestä, tulisi sen tueksi kehittää selkeät ja johdonmukaiset ennakotehtävät, jotka tukevat prosessimallin vaiheita. H5:n mukaan on kuitenkin mietittävä tarkkaan, mille tasolle prosessien määrittäminen jätetään. Yleisesti prosessit aiheuttavat paljon keskustelua ja niiden kanssa pitää osata vetää raja ylätasolle eikä lähteä määrittämään liian yksityiskohtaisesti.

H5:n mukaan esiselvitysvaiheessa tulisi keskustella asiakkaan kanssa millaiset valmiustasot heillä on ja toimiiko projektitiimi liiketoiminnan sparraajana, jolloin budjetti kasvaa vai onko asiakkaalla selkeät raamit toteutettavalle projektille ja haluavat projektitiimin toimimaan vain teknisinä osajina, jolloin budjetti on pienempi. Tämä auttaisi analyysivaiheen räätälöintiä asiakaskohtaisesti ja mahdollistaisi erityyppisten analyysivaiheiden tunnistamisen valmiustason mukaan. H4 puolestaan näkee asiakkaan käyttäjäryhmien osallistamisen tärkeänä asiana, mutta asiakas tulee saada sitoutumaan projektiin tarvittavalla tarkkuudella. Osallistuvien henkilöiden tulee olla sellaisia, jotka pystyvät kertomaan käyttäjäryhmän näkemyksiä sekä haluavat osallistua projektiin. Yleisesti voidaan ajatella, että sitoutunut asiakas takaa paremmin onnistuneen projektin. Schön et al. (2017) näkevät käyttäjien osallistamisen myös tärkeänä ja heidän mukaan erilaiset ketterät menetelmät tukevat käyttäjän sitouttamista projektiin. Samalla ketterät menetelmät tukevat ihmiskeskeistä suunnittelua.

H2 ja H6 näkevät esiselvitysvaiheen järkevänä kokonaisuutena ja kaikki vaiheet tavallaan hyödyllisinä, mutta käytännössä joidenkin vaiheiden yli joudutaan hyppäämään.

Tämä riippuu täysin projektin laajuudesta ja siitä kuinka paljon järjestelmässä on jo olemuksena valmiina. Kaikissa tilanteissa ei välttämättä tarvita esiselvitystä niin paljoa. Kuitenkin erityisesti isojen projektien analyysivaiheen esiselvitysvaihe on tärkeä.

H1, H2, H3, H4 ja H5 näkevät haastattelut tärkeimpänä metodina esiselvitysvaiheessa. Asiakkaan tulee olla koko ajan mukana ja osallistaa heitä mahdollisimman paljon. Erityisesti H6 näkee kaiken yhteisen tekemisen kuten post-it lappujen kanssa työskentelyn ja erilaiset työpajat hyvinä metodeina, joiden aikana saadaan käyttäjiä osallistumaan aktiivisemmin. Kuitenkin suurin osa haastateltavista (H1, H2, H4, H5 ja H6) nostivat esille, että erilaiset palvelumuotoilumetodit tarvitsevat tuekseen materiaaleja, jotka neuvovat selkeästi mitä, asioita tehdään missäkin vaiheessa.

”Jos ei ole selkeää tietoa, millaisia materiaaleja pitää tuottaa, saamme lopputuloksena niin monta erilaista materiaalia kuin on tekijääkin.” (H2)

Erityisesti H4 ja H5 pitävät kaikkia materiaaleja, joita esiselvitysvaiheessa tuotetaan hyvänä. Kaikki materiaalit mitä tuotetaan prosessin alussa, voitaisiin pitää ohjaavina materiaaleina myöhemmissä vaiheissa.

”Esiselvitysvaiheen materiaalit voisivat toimia ohjaavina materiaaleina prosessin myöhemmissä vaiheissa. Voitaisiin tarkistaa, onko painotettu vain niitä kovaäänisiä sidosryhmiä ja heidän vaatimuksiansa ja vahingossa jätetty jotain tärkeitä käyttäjäryhmiä taka-alalle.” (H5)

Mielenkiintoisena näkökulmana H4 nosti esille, että välillä asiakkaan edustajat eivät halua tai uskalla sanoa omia mielipiteitä muiden kuullen ja siksi kyselyt voisivat toimia hyvänä menetelmänä selvittää asiakkaan mieltymyksiä analyysivaiheen alussa. Kyselyn tulisi olla hyvin kevyentason kysely, jonka toteutukseen ei mene paljoa aikaa. Erityisen hyvä olisi, jos kyselyn voisi kevyesti räätälöidä eri käyttäjäryhmien mukaan niin, että talouspuolen henkilöiltä ei kysytä samoja asioita kuin esimerkiksi myynti- tai markkinointihenkilöiltä. Zowghi ja Coulin (2005) sekä Stickdorn (2018) korostavat, että kyselyt toimivat vaatimusmäärittelyn alussa ja ne tulee suunnitella osallistuvien henkilöiden mukaan. Kyselyt auttavat laajentamaan näkökulmia käyttäjäryhmien tarpeista ja kipupisteistä ja tämä näkemys tukee H4:n ajatusta siitä, että kyselyillä voidaan saada esille jotain tärkeitä asioita projektin vaatimusten kannalta.

Esiselvitysvaiheeseen tulee osallistua erityisesti projektin projektipäällikkö ja osaava ratkaisukonsultti. (H1, H2, H3, H6) H4 ja H5 näkevät myös hyvin tärkeäksi sen, että tekninen osaaja kuten tekninen arkkitehti osallistuu jo esiselvitysvaiheessa analyysivaiheeseen. Heidän mielestään tekninen osaaja tulee osallistaa ihmislähtöiseen suunnitteluun samalla tavalla kuin ratkaisukonsultti tai projektipäällikkö. Jotta ratkaisusta saadaan onnistunut, tulee myös teknisen toteuttajan ymmärtää asiakkaan vaatimuksia ja prosesseja. Kaikilla haastateltavilla oli kuitenkin yhteinen näkemys siitä, että asiakkaan puolelta on tärkeää löytää sellaiset henkilöt, jotka tuntevat prosessit hyvin ja pystyvät ottamaan vastuuta siitä, että päätöksiä tehdään heidän mielipiteidensä kautta. Tarvittaessa eri sidosryhmien edustajat käyvät sisäisiä keskusteluita asioista, jotka ovat epäselviä ja esittävät ne erilaisissa tapaamisissa projektitiimille.

6.3.3 Tuotteen kehitysjono

Prosessimallin toinen osa-alue aiheutti paljon epävarmuutta ja pohdintaa haastateltavien keskuudessa. Erityisen hankalana asiana pidettiin sitä, miten vedetään raja vaatimusten ja määrittelyn välille. (H3, H4, H5, H6) Usein analyysivaiheessa lähdetään liikaa määrittämään asioita, vaikka tarkoituksena on vain laajentaa näkemystä ja tunnistaa eri vaatimuksia, mitä asiakkaalla on. Erityisesti mietitystä herätti se, mille tasolle käyttäjätarinat tulee mallintaa ja voitaisiinko kohdeyrityksen ketterässä projektimenetelmässä käsitellä vaatimuksia vielä tarkemmin ennen sprinttiä ja esimerkiksi jakaa isoja vaatimuksia pienempiin osiin. Tässä vaiheessa, kun vaatimuksia selvitetään ja kirjataan ylös, on vaarallista, että sidotaan kädet johonkin tiettyyn määrittelyyn, joka ei myöhemmissä vaiheissa onnistu (H3). H5:n mukaan on hankalaa määrittää rajaa vaatimusten ja määrittelyn välille sekä yrittää määrittää työmääräarvioita vaatimuksille, jotka muuttuvat projektin edetessä.

Yleisesti haastatteluiden perusteella käyttäjätarinoita pidetään hyvänä menetelmänä, kunhan niistä saadaan tarpeeksi tarkkoja kokonaisuuksia. Vaatimusten luokittelu on projektin toteutuksen kannalta tärkeää. Vaikka luokittelut olisivat täysin keksittyjä, tulee sitä silti tehdä, koska se helpottaa vaatimusten käsittelyä. (H5, H6)

H1 näkee, että vaatimusten validointi, dokumentointi ja hallinta jää usein projekteissa taustalle kovan kiireen vuoksi ja keskustelua käydään usein tuotettujen prototyyppien kautta. Prototyypeillä H1 tarkoittaa jo toteutettuja ominaisuuksia, jotka alkavat jollain tavalla jo muodostamaan osia lopullisesta järjestelmästä. H2 puolestaan näkee tilanteen

niin, että kunhan vaatimukset saadaan määritettyä ja priorisoitua minimitoimitukselle, riittää se, eikä muulla ole niin väliä. Vaatimukset, jotka eivät kuulu minimitoimitukseen voidaan hänen mukaansa jättää taka-alalle. H2 ja H6 näkevät onnistumisen mittaamisen hyvänä asiana uudessa prosessimallissa. Kuitenkin niiden toteutuminen riippuu täysin siitä riittääkö aika edes vaatimusten selvittämiseen.

Yleisesti palvelumuotoilumetodit nähdään hyvinä menetelminä selvittää käyttäjäryhmien vaatimuksia sekä mallintaa niitä paremmin. H4, H5 ja H6 näkevät erityisesti työpajat ja erilaiset kaaviot hyvinä työkaluina kehitysjonoa määriteltäessä. Tämä kuitenkin vaatii työpajojen vetäjiltä kykyä ohjata tilannetta ja keskustelua oikeaan suuntaan. Joissain tilanteissa esimerkiksi työpajat tarvitsevat kunnon valmistautumista, jotta työpajasta saadaan toimiva. Zhu (2009) mukaan on myös tärkeää hyödyntää vaatimusmäärittelyn käytäntöjä kuten tarkkailua, haastatteluita, työpajoja sekä vahvaa yhtenäistä tekemistä ketterässä vaatimusmäärittelyssä.

6.3.4 Vaatimusten hallinta

Vaatimusten hallinta tiedostetaan hyvin tärkeänä osana projektia, mutta kiireisen aikataulun takia siihen panostetaan liian vähän. Projekteilla on aina kova kiire ja siksi on hankalaa reagoida muuttuviin tai uusiin vaatimuksiin (H1). Vaatimusten tarkastelu myöhemmissä vaiheissa projekteja voi muuttaa paljon asioita ja tarkastelu auttaisi saamaan enemmän palautetta asiakkailta.

”Välillä projekteissa pitäisi nostaa jalkaa kaasupolkimelta ja katsella uusin silmin vaatimuksia, joita projektin alussa tai ennen projektia on tehty.” (H1)

Sekä H1, H2, H3 ja H4 kritisoivat nykyisen projektimallin ketteryyttä. Usein kiireellinen aikataulu ja kompleksiset kokonaisuudet ajavat projektit siihen tilanteeseen, että toteutuksen aikana ei saada tarpeeksi palautetta asiakkaalta eikä uusiin vaatimuksiin voida reagoida. Vaatimusten hallinta nähdäänkin hyvin hankalana kokonaisuutena sillä se vie paljon aikaa ja harvemmin tähän vaiheeseen on varauduttu tarpeeksi. On hankalaa ottaa projektiin muuttuvia vaatimuksia mukaan, koska aikaisemmin on lupauduttu toteuttamaan projektin minimitoimitus. Se on sopimuksellisesti sidottu ja kaikki muut vaatimukset, joita minimitoimitukseen ei kuulu toteutetaan, jos minimitoimituksen jälkeen jää aikaa. (H2)

”Meidän ketterässä mallissa on paljon ei ketteriä menetelmiä, jotka eivät anna tilaa ketteryydelle. Projektin kokonaisuudesta jopa 80% voi olla minimitoimitukselle kirjattuja asioita. Tällaisessa tilanteessa uusien vaatimusten mukaan nouseminen on mahdollonta.” (H1)

H1, H4 ja H5 näkevät vaatimusten hallinnassa tärkeäksi asiakasi vaatimusten tarkastelun myöhemmissä projektin vaiheissa. Projektien toteutuksen aikana tarvitaan aikaa hengähtää ja ajatella asioita uusin silmin. Tämän takia olisi hyvä, jos analyysivaiheen aikana tunnistettuja vaatimuksia käsiteltäisiin myöhemmin projektin toteutuksen aikana. H5:n ja H6:n mukaan mitä tiukempi analyysivaiheen aikataulu on, sitä vaikeampi vaatimuksia on asettaa oikealle tarkkuustasolle.

6.3.5 Mallin käyttöönotto

Jokainen haastateltava oli sitä mieltä, että malli on toimiva ja erittäin kattava. Käytännössä mallia on kuitenkin hankala toteuttaa kovan kiireen takia. (H1) Erityisen positiivisena asiana nähdään kuitenkin se, että lähdetään liikkeelle käyttäjien tarpeista ja käyttäjäryhmiä otetaan paremmin huomioon. Myös vaatimusten hallinta vie projektin toteutusta oikeaan suuntaan. (H1, H2, H5) Malli jäsentelee hyvin sen, miten analyysivaihetta on toteutettu ja miten sitä pitäisi kehittää eteenpäin. (H6)

Kaikki kartat, kaaviot, yhteistyöpajat ja menetelmät, jotka jollain tavalla selkeyttävät analyysivaihetta nähdään arvokkaina. (H4, H5 ja H6) Käytännön työkalut auttaisivat hahmotamaan analyysivaihetta paremmin ja yleisesti toivotaankin, että olisi olemassa erilaisia konkreettisia työkaluja ja sisäistä kokemusten vaihtoa. (H5) Kuitenkin kaikki menetelmät vaativat tuekseen erilaisia ohjeita ja materiaaleja, jotta jokainen pystyy niitä hyödyntämään.

Jotta analyysivaiheen uusi prosessimalli voitaisiin ottaa käyttöön tulisi asiakas sitouttaa analyysivaiheeseen. Analyysivaihe tulee myydä hyvin asiakkaille, jotta myös asiakas ymmärtää sen arvon. (H5, H6) Yleisesti projekti on helpompi toteuttaa, kun analyysivaihe on toteutettu ja siksi tämä näkemys pitää saada välitettyä myös asiakkaille. Asiakas ei välttämättä näe arvoa analyysivaiheen toteutuksessa ja siksi sitä ei pystytä toteuttamaan suunnitellussa laajuudessa. Joissain tilanteissa voidaan joutua tiivistämään mallia ja toteuttamaan se asiakkaan budjettien puitteissa. (H2)

Ongelmana puolestaan nähtiin se, miten perusteellisesti dokumentoidut ja validoidut vaatimukset istuvat ketterään projektimalliin. (H1) Prosessimallissa tulee jättää tilaa ketteryydelle ja alkuun määrittää isot linjat, joita kokeilemalla voidaan kehittää vaatimuksia. Toisaalta H2:n mukaan vaatimusten muuttaminen ei onnistu sillä, jos niitä halutaan sisällyttää minimitoimitukseen, vaatii se projektin ohjausryhmän hyväksynnän ja harvoin tätä lähdetään toteuttamaan. Nämä näkemykset ovatkin hieman ristiriitaiset keskenään. Toisaalta halutaan, että prosessimalli jättää tilaa ketteryydelle, mutta samalla tiedostetaan, että ketteryys ei tule toteutumaan nykyisen projektimallin takia.

Vaikka työn lähtökohtana on formalisoida analyysivaihetta käyttäjäkeskeisempään suuntaan, nähdään toiminnallisten prosessien määrittäminen silti tärkeänä asiana. (H2, H1) Käyttäjäkeskeisyys siirtää paljon vastuuta asiakkaan puolelle ja tuleeikin olla tarkkana, että asiakas on mukana koko prosessin ajan. Jos asiakas jää pimentoon jossain vaiheessa voi se kostautua projektin myöhemmissä vaiheissa. (H3) Toisaalta ongelmana voidaan nähdä myös aikataulu ja asiakkaan resurssit. (H5) Välttämättä vaadittavia resursseja ei ole saatavilla silloin, kun niitä tarvitaan. Tämä aiheuttaa omia haasteita analyysivaiheen onnistumiselle. Asiakkaan tulee olla aktiivinen analyysivaiheen aikana, jotta päästään parhaaseen lopputulokseen. Jos asiakas ei suostu tekemään esitehtäviä ja olemaan aktiivinen prosessin alkuvaiheessa, lupaa se huonoa koko projektin toteutukselle. Silloin asiakas ei ole tarpeeksi sitoutunut projektiin. (H6)

H4:n mukaan on muistettava, että kun asiakas sitoutuu analyysivaiheen toteutukseen ei se tarkoita sitä, että asiakas on sitoutunut koko projektin toteutukseen. Asiakkaalla on aina mahdollisuus analyysivaiheen jälkeen vaihtaa projektin toimittajaa tai joissain tilanteissa perua koko toteutusprojekti. Tällaisissa tilanteissa ei analyysivaihetta tule luokitella epäonnistuneeksi. Jos lopputuloksena saadaan tilanne, jossa projektia ei kannata toteuttaa on se myös asiakkaan ja toimittajan kannalta hyvä tilanne. Esimerkiksi analyysivaiheen aikana asiakas voi todeta, että maksukyky ei riitä kattamaan projektille asetettua minimitoimitusta ja siksi ei ole kenenkään kannalta järkevää lähteä toteuttamaan itse kehitysprojektiä.

6.4 Ketterän ohjelmistoprojektin vaatimusmäärittelyn viitekehys

Jotta vaatimusmäärittelyn viitekehyksestä saataisiin toimiva, tulee sen rakennetta hieman muuttaa. Haastatteluiden avulla saatujen tulosten pohjalta voidaan kehittää teorian

pohjalta muodostettua viitekehystä sekä arvioida sen toimivuutta. Aikaisemmin työssä on puhuttu paljon prosessimallista, mutta tässä vaiheessa on järkevämpää puhua viitekehuksesta. Kuten haastatteluiden aikana todettiin, jokainen toteutusprojekti on hieman erilainen ja siksi ei voida muodostaa yhtenäistä prosessimallia, joka toimisi jokaisessa tilanteessa. Asiakkaat, toteutettavat projektit ja tilanteet ovat erilaisia ja jos työssä puhutaan prosessimallista, tulisi jokaiselle tilanteelle olla erilainen tilanteeseen sopiva prosessimalli. Tämän vuoksi tutkimuksen tuloksena esitetään yleinen viitekehys, jota voidaan soveltaa eri tilanteisiin.

Haastatteluiden aikana korostui näkemys siitä, että kiireisten aikataulujen takia muodostettua viitekehystä on vaikea toteuttaa sille asetetulla laajuudella. Viitekehys sisältää paljonvaiheita, joita joudutaan joko lyhentämään tai jättämään väliin todellisessa toteutuksessa. Tämän takia viitekehystä voidaan selkeyttää entisestään. Viitekehys voidaan jakaa haastatteluiden tulosten mukaisesti kolmeen päätasoon, joita ovat: esiselvitys, tuotteen kehitysjono ja ylläpito. Tässä vaiheessa kuitenkin selkeyden vuoksi tuotteen kehitysjonovaihe on hyvä nimetä käyttäjätarpeiksi ja ylläpito jatkuvaksi suunnitteluksi ja toteutukseksi. Liitteessä 3 on esitetty uusi viitekehysten rakenne.

Haastatteluissa nousi myös esille, että teorian perusteella muodostettu viitekehys on hyvin selkeä, johdonmukainen ja ymmärrettävä. Viitekehysten esittäminen kolmen päätasoon kautta auttaa kuitenkin paremmin ymmärtämään sitä, millaisia dokumentteja ja tukimateriaaleja missäkin vaiheessa tulee tehdä sekä uusi viitekehys alleviivaa esiselvityksen tärkeyttä. Viitekehysten vaiheet tarvitsevat tuekseen vielä enemmän konkretiaa sekä ohjeita siitä, miten erilaisia palvelumuotoilun metodeja hyödynnetään. Haastatteluissa kyseenalaistettiin myös aikaisemman prosessimallin vaiheita: vaatimusten selvittäminen, analysointi, dokumentointi ja validointi. Koska viitekehysten tulee toimia ketterille ohjelmistoprojekteillemme, rajoittavat näin yksityiskohtaiset vaiheet ketteryyttä. Tämän takia aikaisemmin esitettyä viitekehystä kehitettiin eteenpäin niin, että prosessivaiheiden 3-6 painoarvoa hieman muutettiin ja esimerkiksi erillinen vaihe dokumentoinnille poistettiin.

Haastatteluiden aikana vahvistui näkemys siitä, että vaatimusmäärittelyn jokaiseen vaiheeseen tulee osallistua ydinprojektitiimi, joka koostuu kohdeyrityksen eli toimittajan asiantuntijoista. Ydinprojektitiimi koostuu kolmesta henkilöstä, joita ovat projektipäällikkö, ratkaisukonsultti ja tekninen arkkitehti. Ydinprojektitiimiä tulee laajentaa projektin vaiheiden tarpeiden mukaisesti erilaisilla osaajilla. Tämä mahdollistaa sen, että projektin eri vaiheissa saadaan tarvittavaa tietämystä muuttuvien tilanteiden mukaan.

Ydinprojektitiimiin kuuluu yleisesti projektipäällikkö, fasilitointivastaava, palvelumuotoilu-vastaava, tekninen vetäjä, palvelumuotoilun hallinnointivastaava, muutosvastaava, liike-toimintavastaava, projektin omistaja ja tuoteomistaja. (Collier, 2011, Stickdorn et al. 2018, ss. 342-344) Haastatteluiden kautta nousi kuitenkin esille se, että kohdeyrityksen projekteissa kolmen henkilön ydintiimi riittää toteuttamaan analyysivaiheen onnistuneesti ja siksi ydinprojektitiimin rooleja on useita yhdellä henkilöllä. Esimerkiksi projektipäällikkö toimii myös fasilitointivastaavana, palvelumuotoilun hallinnointivastaavana, projektin omistajana sekä tuoteomistajana. Liitteessä 4 on esitetty viitekehys, jota on rikastettu osallistuvien henkilöiden sekä palvelumuotoilumetodien avulla.

Asiakkaan puolelta ydinprojektitiimi vaihtelee prosessivaiheiden mukaan. Esiselvitysvaiheessa, kun määritetään reunaehdot, asiakkaalta osallistuu liiketoimintayksiköiden vetäjät, joilla on mahdollisuus tehdä ylätasoa päätöksiä. Käyttäjien ja sidosryhmien tunnistamisen jälkeen projektitiimi laajenee myös kaikkiin tärkeisiin sidosryhmien edustajiin ja heitä osallistetaan projektiin. Tässä vaiheessa on hyvä, että asiakkaan puolelta osallistuu tärkeäksi tunnistetut käyttäjät ja sidosryhmät sekä henkilöt, jotka pystyvät tekemään ylätasoa päätöksiä. Asiakkaan tulee aina nimetä omalle projektilleen tuoteomistaja, jonka vastuulla on pitää huoli siitä, että kaikki asiat ovat eri liiketoimintayksiköiden edustajien mukaisia ja tuoteomistajan tulee pystyä tekemään tärkeitä päätöksiä projektin aikana.

Esiselvitys

Esiselvitysvaiheen tavoitteena on määrittää liiketoimintatavoitteet ja -ongelmat, joihin halutaan ratkaisuja uuden järjestelmän myötä. Esiselvityksen aikana määritetään toteutettavan projektin reunaehdot sekä järjestelmän kehitys- ja käyttöympäristö. Tässä vaiheessa määritetään projektin aikataulu ja laajuus, hyödynnettävät teknologiat sekä lähejärjestelmät ja mahdolliset aikaisemmat ratkaisut. Käyttöympäristöstä on puolestaan hyvä tietää käyttäjien määrää ja suhteet muihin järjestelmiin. Esiselvitysvaiheessa on tärkeää arvioida sitä, millaisia liiketoimintatavoitteita asiakkaalla on ja mitä liiketoiminnan osa-alueita tarvitaan projektiin mukaan. Samalla voidaan rajata joitain alueita myös pois projektin laajuudesta. Liiketoimintaongelmien ja -tavoitteiden määrittämisessä on hyvä miettiä asiakkaan historiaa ja nykytilaa.

Esiselvitysvaiheessa projektipäällikkö kutsuu koolle ydinprojektitiimin sekä asiakkaan puolelta liiketoimintayksiköiden edustajat sekä henkilöt, joihin projekti vaikuttaa. Projekti-

päällikkö fasilitoi yhteistyöpajatilaisuuden, jossa käydään läpi projektin reunaehdot. Työpajan aikana käytetään palvelumuotoilumetodeista haastatteluista keskustelun ylläpitämiseen. Yhteistyöpajassa hyödynnetään puolistrukturoitua haastattelua, jota voidaan tukea 5 x Miksi -metodin tai kohderyhmähaastattelun keinoin. Esiselvitysvaiheessa hyödynnetään tukidokumenttina kysymys- ja vastauspohjaa, johon haetaan työpajan aikana vastauksia liiketoiminnan ongelmiin ja tavoitteisiin, nykytilaan sekä kehitys- ja käyttöympäristöön. Kysymyslista voidaan lähettää etukäteen sähköpostitse asiakkaalta osallistuville henkilöille. Lisäksi esiselvitysvaiheessa tarvitaan tukidokumenttina työpajan järjestämisen ohjeet sekä palvelumuotoilumetodien käyttöohjeet.

Käyttäjätarpeet

Viitekehysten toisessa vaiheessa määritetään tunnistettujen käyttäjien ja sidosryhmien roolit sekä käyttäjätarpeet. Käyttäjätarpeiden tulee liittyä esiselvitysvaiheessa asetettuihin liiketoiminnallisiin tavoitteisiin. Tässä vaiheessa tarpeita tulee käsitellä käyttäjäroolien kautta yhteistyöpajoissa. Yhteistyöpajojen määrä riippuu erilaisten käyttäjäroolien määrästä ja vaihtelee toteutettavan projektin mukaan. Viitekehys ei ota kantaa siihen, kuinka monta yhteistyöpajaa tulee pitää, jotta jokainen käyttäjärooli on otettu huomioon.

Kun käyttäjätarpeita määritetään, tulee yrittää ymmärtää käyttäjän toimintaa, kun käyttäjä käyttää järjestelmää. Järjestelmän tulisi aina pystyä tukemaan käyttäjän toimintaa ja tarpeita mahdollisimman hyvin. Kaikkia tarpeita ei kuitenkaan pystytä määrittämään etukäteen tarkasti, koska käyttäjien ymmärrys voi olla hyvin vähäistä projektin alkuvaiheessa. Käyttäjien ymmärrys kehittyy projektin aikana ja siksi uusia tarpeita voi muodostua myös myöhemmin. Myös muutokset liiketoimintaympäristössä kuten vaihtuva henkilöstö, organisaatiouudistukset ja teknologiat aiheuttavat muutoksia liiketoimintatavoitteissa ja samalla myös käyttäjätarpeissa. Tämän takia ei ole järkevää määrittää käyttäjävaatimuksia liian tarkasti, sillä tilanteet muuttuvat ketterän ohjelmistoprojektin toteutusvaiheessa.

Haastatteluiden aikana kyseenalaistettiin tarkkuustasoa, jolla vaatimuksia tulisi määrittää. Pohdintaa herätti myös se, koska tiedetään, onko vaatimukset määritetty riittävän tarkalle tasolle, mutta ei kuitenkaan niin tarkasti, että olisi menty määrittelyn puolelle. Ketterälle kehitykselle on tyypillistä, että vaatimuksia ei dokumentoida liian aikaisessa vaiheessa ja kehitysjono muodostetaan vai käyttäjätarinoiden sekä priorisoinnin kautta. (Cohn, 2004; Collier, 2011) Käyttäjätarinat voidaan jakaa kolmeen päätasoon: teemat, epiikat ja käyttäjätarinat (Girvan & Paul, 2017). Tämä ajattelu auttaa analyysivaihetta ja

se tulee ottaa huomioon viitekehyksen yhteydessä. Muodostetun viitekehyksen käyttäjätarpeiden vaiheessa tunnistetaan ensin käyttäjä- ja sidosryhmät, jonka jälkeen heidän tavoitteita ja tehtäviä käydään läpi. Tässä vaiheessa käyttäjätarinoista muodostetaan teemoja, jotka toimivat ylätasoina käsitteinä eri käyttäjätarinaroryhmille. Girvanin ja Paulin (2017) mukaan teemat pyrkivät aina olemaan ylätasoina kuvauksia ja niiden tulisi kuvata käyttäjien tärkeimpiä tavoitteita. Teemoja ei tule vielä kirjoittaa käyttäjätarinoiden muotoon. Kun siirrytään käyttäjien tavoitteista ja tehtävistä viitekehyksessä analyysivaiheeseen pyritään tunnistamaan kaikki käyttäjien ja sidosryhmien vaatimukset. Tässä vaiheessa selkeimmät vaatimukset voidaan kirjoittaa jo käyttäjätarinoiksi, mutta kaikki laajemmat käyttäjätarinat nimetään epiikoiden mukaan. Epiikat tulee jaotella pienempiin kokonaisuuksiin, joko analyysivaiheessa, jos se on mahdollista tai sitten myöhemmin hanketasolla, kun vaatimuksia otetaan kehitysjonolta tarkempaan käsittelyyn.

Käyttäjätarpeiden määrittäminen tulisi tapahtua niin, että sen perusteella pystytään antamaan luotettava arvio toteutuksen kustannuksista, mutta samalla tulee hyväksyä ketterän toteutuksen periaatteet ja se, että kaikkea ei pystytä tunnistamaan ja määrittämään etukäteen. Vaatimukset tulisi myös tarkistaa aina ennen toteutusta ja siirtää tuotteen kehitysjonolta vaatimuksia sprintin kehitysjonolle. Jos vaatimukset muuttuvat tai esille nousee täysin uusia vaatimuksia, tulee asiakkaan aina hyväksyä ne. Usein muuttuvat vaatimukset muuttavat projektin kustannuksia ja laajuutta.

Kun määritetään tuotteen kehitysjonoa, on erittäin tärkeää, että käyttäjät ja sidosryhmien edustajat ovat koko ajan läsnä toteutuksessa. Koko vaatimusmäärittely on iteratiivinen prosessi, joten kaikki projektin aikana esille nousseet asiat vaikuttavat koko projektin etenemiseen.

Käyttäjätarpeenvaiheessa palvelumuotoilumetodeista erityisesti yhteistyöpajat ovat hyödyllisiä. Kun halutaan ymmärtää paremmin käyttäjien vaatimuksia, voidaan niihin syventyä teemahaastatteluiden avulla sekä samalla voidaan tutustua olemassa olevaan ratkaisuun. Käyttäjätarpeita puretaan käyttäjäroolien ja erityisesti käyttäjätarinoiden avulla. Käyttäjäroolien kuvailemisessa voidaan laatia erilaisia persoonakuvauksia ja sidosryhmäkartoja sekä niiden avulla priorisoida eri käyttäjäryhmiä. Myös tietomallinnusta voidaan tehdä yhteistyöpajojen aikana esimerkiksi korttien lajittelun avulla. 5 x Miksi -tekniikka puolestaan auttaa vaatimusten juurisyiden selvittämisessä.

Jatkuva suunnittelu ja toteutus

Viitekehyyksen kaksi ensimmäistä vaihetta toimivat jo itsessään tuottamaan projektin alussa käyttäjävaatimukset sekä luomaan jollain tavalla toteutuskelpoisen tuotteen kehitysjonon. Kuitenkin, koska kyseessä on ketterä ohjelmistoprojekti ja sen analyysivaihe, tulee myös vaatimusmäärittelyssä miettiä, miten muuttuvat vaatimukset otetaan huomioon projektin toteutuksen aikana sekä miten aikaisemmassa vaiheessa määritellyt epäikat saadaan jaoteltua pienempiin käsiteltäviin osiin.

Jotta muuttuvia tai uusia esille nousseita vaatimuksia voidaan käsitellä projektitoteutuksen aikana, tulee projektitiimin pyrkiä tiheisiin julkaisuihin. Vaatimukset tulee jakaa pienempiin osiin lähempänä toteutettavia sprinttejä, jotta projektin toteutus tukee vaatimusten hallintaa ja tiheitä julkaisuja. Säännölliset ja tiheät julkaisut mahdollistavat haluttujen toteutusten julkaisun nopeammin sekä niiden avulla mahdollistetaan palautteen saaminen asiakkaalta. Joissain tilanteissa nopeasti toteutettuja toimivia ominaisuuksia voidaan pitää järjestelmän prototyyppeinä ja niiden kautta asiakkaan tietoisuus kasvaa.

Jotta jatkuvaa suunnittelua ja toteutusta voidaan toteuttaa koko ohjelmistoprojektin toteutuksen ajan, tulee minimivaatimuksen laajuutta pienentää sekä antaa projektien toteutuksen aikana tilaa ketteryydelle. Toteutuksen aikana tulee myös peilata toteutuneita vaatimuksia aikaisemmissa vaiheissa toteutettuihin materiaaleihin. Projektipäällikön tulee kriittisesti tarkastella, onko kaikkien sidosryhmien vaatimukset edelleen otettu tarpeeksi tarkasti huomioon, eikä jonkin käyttäjän tai sidosryhmän vaatimuksia ole jätetty taka-alalle. Samalla toteutuksen aikana projektitiimin on hyvä kerätä palautetta käyttäjäryhmiltä ja pyrkiä jatkuvasti parantamaan toimintaa.

7. YHTEENVETO JA PÄÄTELMÄT

Tutkimuksen viimeisessä kappaleessa vastataan tutkimuksen alussa esitettyihin tutkimuskysymyksiin ja keskustellaan yleisesti tutkimuksen tuloksista. Tutkimuksen tuloksia arvioidaan kriittisesti sekä nostetaan esille erilaisia jatkotutkimusmahdollisuuksia, joita tutkimuksen toteutuksen aikana on tunnistettu ja joilla voidaan syventää tämän tutkimuksen tuloksia.

7.1 Tulosten yhteenveto ja vastaukset tutkimuskysymyksiin

Tutkimuksen tarkoituksena oli tutustua kohdeyrityksen ketterien projektien analyysivaiheeseen sekä kehittää analyysivaihetta käyttäjäkeskeisempään suuntaan. Aikaisemmin vaatimusmäärittelyä on lähdetty toteuttamaan asiakkaan toiminnallisten prosessien kautta. Kohdeyrityksessä on kuitenkin tunnistettu tarve kehittää analyysivaihetta ihmiskeskeisempään suuntaan niin, että toiminnallisia vaatimuksia mallinnetaan käyttäjien ja sidosryhmien avulla.

Työn tarkoituksena oli tutustua ketterän ohjelmistoprojektin vaatimusmäärittelyyn sekä muodostaa viitekehys kriittisen kirjallisuuskatsauksen kautta käyttäjä- ja ihmiskeskeiseen vaatimusmäärittelyyn. Viitekehystä rikastettiin erilaisilla palvelumuotoilumetodeilla, jotka tukevat ketterää vaatimusmäärittelyä. Tutkimuksen tavoitteena oli luoda ihmiskeskeytyksen perustuva viitekehys, jonka lopputuloksena syntyy projektin käyttäjätarinat eli toiminnalliset vaatimukset. Viitekehysten tarkoituksena on mahdollistaa käyttäjätarpeiden määrittäminen oikealla tarkkuustasolla.

Viitekehykselle asetettiin 3 erilaista tavoitetta: Sen tulee 1. Tukea ketterän ohjelmistoprojektin vaatimusmäärittelyä sekä toimia valmishjelmistojen toteutusprojekteissa, 2. Tukea eri osapuolien välistä kommunikointia sekä varmistaa, että kaikki vaatimukset saadaan mallinnettua oikealla tarkkuustasolla oikeilta henkilöiltä ja oikeasta liiketoiminnan näkökulmasta, 3. Mahdollistaa toiminnallisten vaatimusten keräämisen käyttäjätarinoina. Viitekehysten tulee tukea käyttäjän tavoitteita ja tarpeita sekä helpottaa projektitiimien työskentelyä jatkossa.

Tutkimus täydentää hyvin kattavasti lähes puuttuvaa empiiristä tutkimusta palvelumuotoilumetodien yhdistämisestä käyttäjäkeskeiseen suunnitteluun ja ketterän projektin vaatimusmäärittelyyn. Aikaisempaa tutkimusta on vain vähän saatavilla käyttäjäkeskeisen suunnittelun yhdistämisestä ketterän ohjelmistoprojektin toteutukseen ja siksi työ täydentääkin erityisesti tätä tutkimusalaa. Tutkimus toteutettiin konstruktivisena tutkimuksena, jonka tarkoituksena oli ensin tehdä laaja kirjallisuuskatsaus, jonka pohjalta muodostettiin konstruktio eli prosessimalli analyysivaiheen toteutuksesta. Tämän jälkeen prosessimallin pohjalta laadittiin empiirisen tutkimuksen haastattelurunko. Empiirisessä tutkimuksessa haastateltiin kohdeyrityksen kuutta eri asiantuntijaa. Haastattelut toteutettiin puolistrukturoituina teemahaastatteluina ja niiden analysointi tapahtui temaattista analyysiä hyödyntäen. Haastatteluiden tulosten pohjalta prosessimallia kehitettiin eteenpäin ja lopulta muodostettiin viitekehys toiminnallisten vaatimusten määrittämiseen käyttäjäkeskeisesti. Muodostettu viitekehys vastaa tutkimuksen päätutkimuskysymykseen.

Tutkimukselle tunnistettiin alussa kaksi alatutkimuskysymystä, joihin vastaamalla pystyttiin vastaamaan tutkimuksen päätutkimuskysymykseen. Alatutkimuskysymyksistä ensimmäinen oli *"Minkälaisen viitekehysten avulla voidaan kuvata toiminnalliset vaatimukset?"* Ketterän ohjelmistoprojektin toiminnallisten vaatimusten määrittämisessä tunnistettiin olevan kolme vaihetta: 1. esiselvitys, 2. käyttäjätarpeet ja 3. jatkuva suunnittelu ja toteutus. Jokaiselle vaiheelle on tunnistettavissa omat tavoitteet, jotka tukevat myös viitekehysten tavoitteiden saavuttamista. Esiselvitysvaiheessa tavoitteena on määrittää projektin reunaehdot, joiden mukaan koko projekti tullaan toteuttamaan sekä tunnistaa ja määrittää käyttök konteksti. Tässä vaiheessa määritetään myös liiketoimintatavoitteet ja -ongelmat, joita toteutettavalla järjestelmällä pyritään ratkaisemaan. Tämän lisäksi esiselvitysvaiheessa tunnistetaan tärkeimmät käyttäjät ja sidosryhmät sekä analysoidaan niiden ominaisuuksia, tavoitteita ja tehtäviä. Esiselvitysvaihe antaa projektille tahtotilan, johon voidaan aina palata projektin toteutuksen aikana. Käyttäjätarpeet vaiheessa selvitetään käyttäjäryhmien vaatimuksia, kirjataan ne käyttäjätarinoina sekä analysoidaan ja validoidaan vaatimukset. Käyttäjäryhmien tarpeiden tunnistamisen lisäksi on tärkeää hyväksyttää kaikki tunnistetut vaatimukset käyttäjien ja sidosryhmien edustajien kanssa. Viitekehysten viimeinen vaihe eli jatkuva suunnittelu ja toteutus sisältää kaiken mikä liittyy vaatimusten hallintaan. Ketterälle projektille on tyypillistä, että kaikkia vaatimuksia ei tule vielä analyysivaiheessa tunnistaa ja käsitellä, vaan itse projektitoteutuksen aikana annetaan tilaa muuttuville ja uusille vaatimuksille. Toteutuksen aikana on tärkeää ottaa huomioon muuttuneet vaatimukset ja tarvittaessa jakaa isoja vaikeasti käsiteltäviä vaatimuksia pienempiin osiin.

Tutkimuksen toinen alatutkimuskysymys oli ”*Miten palvelumuotoilua voidaan hyödyntää toiminnallisten vaatimusten määrittelyssä?*” Palvelumuotoilumetodit tukevat toiminnallisten vaatimusten määrittämistä erityisesti esiselvitys- ja käyttäjätarvevaiheissa. Palvelumuotoilumetodit tukevat kaikkia viitekehykselle asetettuja tavoitteita. Metodit tukevat ketterän ohjelmistoprojektin vaatimusmäärittelyä. Ketterän ohjelmistoprojektin vaatimusmäärittely voidaan nähdä hyvin hankalana ja kompleksisena kokonaisuutena ja siksi palvelumuotoilumetodit selventävät sitä, mitä pitää tehdä, jotta vaatimuksia voidaan tunnistaa käyttäjäkeskeisesti. Palvelumuotoilumetodit tukevat käyttäjäkeskeisyyttä osallistamalla käyttäjää ja samalla mahdollistavat myös hankalien kokonaisuuksien käsittelyn.

Viitekehyksen toinen tavoite on tukea eri osapuolien välistä kommunikointia sekä varmistaa, että kaikki vaatimukset saadaan mallinnettua oikealla tarkkuustasolla oikeilta henkilöiltä ja oikeasta liiketoiminnan näkökulmasta. Palvelumuotoilumetodit tukevat näitä kaikkia tavoitteita ja ottavat tavoitteet huomioon koko prosessin ajan. Oikeiden henkilöiden mukaan ottamista tukee ensinnäkin viitekehyksen käyttäjäkeskeinen ajattelu ja palvelumuotoilumetodeista sidosryhmäkartat, persoonakuvaukset sekä haastattelut auttavat tunnistamaan oikeat henkilöt. Samalla metodien avulla mahdollistetaan tehokas tiedon kerääminen. Erilaiset metodit tukevat myös eri osapuolien välistä kommunikointia, sillä puoliavoimet haastattelut, yhteistyöpajat ja esimerkiksi skenaariotyöskentely vaativat kommunikointia käyttäjien ja projektitiimin välillä. Palvelumuotoilumetodien tarkoituksena on osallistaa eri ihmisiä yhteiseen tekemiseen.

Viitekehykselle asetettu viimeinen tavoite on mahdollistaa toiminnallisten vaatimusten keräämisen käyttäjätarinoina. Tätä tavoitetta tukee erityisesti palvelumuotoilumetodeista yhteistyöpajat ja itse käyttäjätarinat. Käyttäjätarinat voidaan luokitella itsessään jo palvelumuotoilumetodiksi ja siksi niitä hyödyntämällä päästään viitekehyksen tavoitteeseen.

Tutkimuksen päätutkimuskysymykseksi asetettiin ”*Miten valmisohjelmiston toiminnalliset vaatimukset määritellään käyttäjälähtöisesti?*” Ketterän ohjelmistoprojektin vaatimusmäärittelylle on tunnistettavissa kolmivaiheinen prosessi, jolla jokaisella vaiheella on omia tavoitteita. Kuten aikaisemmin todettiin näitä prosessin vaiheita ovat esiselvitys, käyttäjätarpeet sekä jatkuva suunnittelu ja toteutus. Prosessi on iteratiivinen ja joustava kokonaisuus, jonka keskiössä toimii erilaiset palvelumuotoilumetodit. Jotta vaatimusmäärittelystä saadaan käyttäjälähtöistä, tulee eri käyttäjät ja sidosryhmät ottaa mukaan toteutukseen varhaisessa vaiheessa. Onnistunut vaatimusmäärittely tarvitseekin useiden henkilöiden osallistumista, jotta viitekehyksen tavoitteet saavutetaan onnistuneesti.

7.2 Tutkimuksen arviointi ja uutuusarvo

Tutkimus toteutettiin mahdollisimman objektiivisesti ja puolueettomasti. Kuitenkin tutkimus sisältää paljon epäluotettavia tekijöitä, jotka vaikuttavat lopputuloksiin. Tutkimusta voidaan arvioida neljän eri tekijän kautta: tutkimuksen luotettavuus ja uskottavuus, tutkimuksen siirrettävyys ja sovellettavuus, tutkimuksen varmuus sekä tutkimuksen neutraalius ja varmennettavuus. (Guba et al. 1994; Whitemore et al. 2001; Thomas & Magilvy, 2011).

1) Tutkimuksen luotettavuus ja uskottavuus

Tutkimuksen tekijällä oli koko tutkimuksen teon ajan työsuhde kohdeyritykseen ja se vaikuttaa jo itsessään tutkimuksen luotettavuuteen ja uskottavuuteen. Työsuhteen merkitys korostui erityisesti haastattelutilanteissa, koska tutkijalla ei ollut aikaisempaa kokemusta kyseisistä työtehtävistä ennen työsuhteen alkua. Jotkut haastateltavat suhtautuivat osin kouluttavaisesti haastatteluihin, joka osaltaan vaikutti haastattelutilanteeseen. Välillä myös haastattelijalle välittyi tuntemus siitä, että kaikki haastateltavat eivät ottaneet hänen ajatuksiaan tosissaan tietäen juuri vähäisen osaamisen aihealueesta.

Jokaiselle tutkimuksen haastatteluun osallistuvalla henkilöllä annettiin mahdollisuus kieltäytyä ja kaikki osallistuivat haastatteluihin vapaaehtoisesti. Tutkimuksen luotettavuutta lisäsi myös se, että jokaisen haastattelun alussa käytiin keskeiset asiat läpi, jotta jokaisella haastateltavalla oli samat lähtökohdat. Teemahaastatteluissa tutkija teki päätökset siitä, mihin haastateltavan kommenttiin tartuttiin. Tämä vaikutti siihen, että joihinkin näkökulmiin on saatettu kiinnittää enemmän huomiota ja samalla tavalla joitain aihealueita on saatettu jättää taka-alalle. Tämä on voinut vaikuttaa suoraan tutkimuksen tuloksiin.

2) Tutkimuksen siirrettävyys ja sovellettavuus

Tutkimus on toteutettu ja kirjoitettu auki niin tarkasti, että se on mahdollista siirtää ja toistaa eri organisaatiossa. Aineiston analysointi on laadittu läpinäkyvästi ja esitetty niin tarkasti, että se tukee siirrettävyyttä. Tutkimus on kuitenkin luottamuksellinen ja siksi haastatteluiden litterointia tai äänitteitä ei voida jakaa julkisesti ja siksi se heikentää tutkimuksen siirrettävyyttä.

Tutkimukseen valitut haastateltavat ja heidän näkemyksensä ovat vaikuttaneet vahvasti tutkimuksen tuloksien muodostumiseen. Tämän takia on epätodennäköistä, että tätä sa-

maa tutkimusta pystyttäisiin toteuttamaan samoilla tuloksilla jossain toisessa kontekstissa. Jos tätä tutkimusta sovelletaan jossain toisessa yrityksessä toisen kohdeyrityksen asiantuntijoille, tulisi sen kautta esille varmasti uusia näkökulmia tutkimuksen tuloksiin.

3) Tutkimuksen varmuus

Suurinta epävarmuutta tutkimukselle muodostaa empiirinen tutkimus. Tutkimuksen otanta on hyvin rajallinen diplomityön aikataulujen vuoksi. Empiirisessä tutkimuksessa haluttiin haastatella henkilöitä, joilla on kokemusta kohdeyrityksessä toteutetuista analyysivaiheista. Koska analyysivaihetta on toteutettu vasta muutamassa projektissa ja se on ollut olemassa hyvin vähän aikaa, rajasi se tutkimuksen otantaa huomattavasti. Tutkimuksen varmuutta olisi lisännyt se, jos viitekehystä olisi pystynyt testaamaan käytännössä. Tämä olisi mahdollistanut sen, että olisi pystytty testaamaan miten malli oikeasti toimii ja miten asiakkaan puolella suhtauduttaisiin siihen, että analyysivaihe lähtee liikkeelle vahvasti käyttäjistä ja sidosryhmistä. Kuitenkin jo tällä tutkimuksen toteutuksella ja otannalla saatiin hyviä tuloksia aikaan, joiden perusteella viitekehystä pystyttiin iteroimaan eteenpäin.

4) Tutkimuksen neutraalius ja varmennettavuus

Tutkimuksen rakennetta on suunniteltu kohdeyrityksen ohjaajien ja tutkimuksen ohjaajan kanssa. Kuitenkin rakenteeseen on vaikuttanut paljon tutkimuksen tekijän omat näkemykset aiheesta ja niiden avulla on täydennetty tutkimuksen rakennetta sille sopivaan muotoon. Tämän avulla varmistettiin se, että tutkimuksen rakenne on tarkoituksen mukainen ja se on linjassa tutkimuksen tavoitteiden kanssa. Myös tutkimuksen tuloksia on pilotoitu konstruktivisen tutkimuksen mukaisesti.

Tutkimuksella onnistuttiin vastaamaan konstruktivisen tutkimuksen tavoitteeseen eli tuottamaan uutta tietoa tutkimuskentälle. Tutkimustulosten luotettavuutta lisää tutkimuksessa muodostettu konstruktio teoriasta ja empiriasta. Konstruktio on luotu pohjautuen teoriaan ja empiriaan, mutta sen luomisessa on myös käytetty paljon tutkijan subjektiivisiä tulkintoja aineistosta.

Tutkimuksen uutuusarvo

Tutkimuksen uutuusarvo on hyvin merkittävä. Tutkimuskentällä ei ole aiempaa tutkimusta, joka yhdistäisi palvelumuotoilumetodit ketterään ohjelmistokehitykseen ja käyttäjätarvemäärittelyprosessiin niin laajasti, kuin tässä tutkimuksessa on tehty. Tutkimuk-

sessä toteutettiin laaja kirjallisuuskatsaus, joka yhdisteli useita erilaisia lähteitä vaatimusmäärittelyn, käyttäjätarpeiden ja ihmiskeskeisen suunnittelun tutkimuksista ja samalla niiden avulla on luotu ymmärrystä eri tutkimusten yhtymäkohdista.

Empiirisen tutkimuksen avulla pystyttiin todentamaan nämä yhtymäkohdat ja niiden merkitys. Tutkimuksessa tunnistettiin se, kuinka vähäistä ketterän ohjelmistoprojektin vaatimusmäärittelyn tutkimus on ja kuinka vähän aikaisempiin tutkimuksiin on liitetty käyttäjä- tai ihmiskeskeisen suunnittelun periaatteita. Toisaalta tutkimuksessa tunnistettiin se, kuinka on olemassa lähes kokonainen tutkimusaukko palvelumuotoilumetodien hyödyntämisessä vaatimusmäärittelyssä. Tutkimuksen kautta toteutettu viitekehys tarjoaa ratkaisun, jota voidaan lähteä testaamaan tulevien projektien analyysivaiheissa.

Empiirisen tutkimuksen kautta nousi myös esille ongelma ketterän kehityksen ideologian ja käytännön toteutuksen välillä. Ketteryys vaatii myös asiakkaalta paljon ja siksi ei ole itsestään selvyyttä, että projekti voidaan toteuttaa ketterän ideologian mukaisesti. Tutkimuksen aihepiiri aiheuttaa myös paljon keskustelua vaatimusten määrittämisestä ja siitä, millainen valmiustaso asiakkaalla on. On hankalaa tietää mille tasolle vaatimukset tulee määrittää ja missä vaiheessa tiedetään vetää raja vaatimuksille ja määrittämiselle. Ketterälle kehitykselle on ominaista, että vaatimukset muuttuvat projektin edetessä, mutta silti analyysivaiheessa pitäisi jollain tasolla tietää, millä tarkkuudella vaatimuksia tulee määrittää, jotta jotain vaatimuksia voidaan lähteä toteuttamaan projektin alkuvaiheissa.

7.3 Tulevaisuuden tutkimuskohteet

Tutkimuksen aihealue on hyvin laaja ja tutkimuksen rajauksilla jätettiin paljon asioita käsittelemättä. Tulevaisuudessa analyysivaiheen viitekehys tarvitsee tuekseen selkeät ohjeet siitä, millaisia esitehtäviä tarvitaan ja millaisia valmiita pohjia voidaan hyödyntää esitehtäviä laatiessa. Palvelumuotoilumetodit tarvitsevat myös tukimateriaalia, jotta jokainen metodi toteutetaan samalla tavalla ja kaikki materiaalit ovat selkeitä ja johdonmukaisia.

Luonnollisesti seuraava jatkotutkimusaihe tutkimukselle on analyysivaiheen viitekehysten testaaminen käytännössä. Viitekehysten pohjalta toteutettu analyysivaihe antaa paljon tietoa mallin toimimisesta käytännössä ja sitä kautta saadun kokemuksen ja palautteen perusteella viitekehystä on mahdollista kehittää entistä toimivammaksi. Kaikki ohjelmistoprojektit ja asiakkaat ovat erilaisia ja siksi toimiva viitekehys tarvitsee tuekseen käytännön tutkimusta ennen, kun malli on toimiva. Haastatteluiden aikana huomattiin,

että riippuen asiakkaasta analyysivaiheen toteutus voi olla hyvin erilainen. Esiselvitysvaiheessa olisi hyvä selvittää asiakkaan valmiustasosta ja projektitiimin roolista. Joissain tilanteissa asiakas haluaa projektitiimin ottavan enemmän vastuuta ja heidän tulee ohjata analyysivaihetta enemmän. Joissain tilanteissa taas asiakas on hyvin itseohjautuva ja tarvitsee vain vähän projektitiimin tukea. Tämän takia analyysivaiheen viitekehystä voisi jatkossa kehittää eteenpäin niin, että riippuen asiakkaan valmiustasosta voitaisiin viitekehystä räätälöidä tilanteeseen sopivaksi. Esimerkiksi voidaan tunnistaa erilaisia toteutustilanteita, joille jokaiselle voidaan viitekehyksen pohjalta muodostaa kuhunkin tilanteeseen sopiva prosessimalli.

Tutkimuksen aikana tuotettu viitekehys tarvitsee tuekseen toimivan ketterän ohjelmistoprojektin, jotta viitekehystä voidaan hyödyntää onnistuneesti. Empiirisen tutkimuksen kautta tunnistettiin, että kohdeyrityksen projektitoteutukset rajoittavat osittain ketteryyttä eikä se tällä hetkellä tue tarpeeksi hyvin käyttäjäkeskeisen vaatimusmäärittelyn piirteitä. Hyvänä jatkotutkimuksena voidaankin pitää projektitoteutuksen muuttamista käyttäjäkeskeisempään suuntaan niin, että kohdeyrityksen projektitoteutus tukee sekä käyttäjäkeskeisyyttä, että ketteriä menetelmiä.

LÄHTEET

Abdulhalim, H., Lurie, Y., Mark, S. (2018). Ethics as a Quality Driver in Agile Software Projects. *Journal of Service Science & Management*, 11, 13.

Agile Alliance. (2001). The Agile Manifesto. Viitattu 18.6.2019, osoitteesta <http://agile-manifesto.org/>

Ardito, C., Baldassarre, M., Caivano, D., Lanzilotti, R (2017) Integrating a SCRUM-based process with Human Centered Design: an Experience from an Action Research Study. *IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*.

Ashraf, S., Aftab, S. (2017). Scrum: An Improved Scrum Process Model. *International Journal of Modern Education & Computer Science; Hong Kong*, 9(8), 16.

Beyer, H., Holtzblatt, K., Baker, L. (2010) An Agile Customer-Centered Method: Rapid Contextual Design, teoksessa. doi: 10.1007/978-3-540-27777-4_6.

Beynon.Davies, P., Sauer, C., Galliers, R. (2009) *Business Information Systems. International Bibliography of the Social Sciences (IBSS)*.

Cao, L., Ramesh, B. (2008) *Agile Requirements Engineering Practices: an Empirical Study*. IEEE Computer Society.

Carrizo, D., Dieste, O., Juristo, N. (2008) "Study of elicitation techniques adequacy", teoksessa WER08 XI Workshop em Engenharia de Requisitos. doi: 10.1016/j.jaci.2012.06.024.

Chesnut, D., Nichols, K. (2014) *UX for dummies*. O'Reilly Safari Learning Platform: Academic edition. 1st ed., John Wiley & Sons, West Sussex, England.

Chua, B.B, Bernardo, D.V., Verner, J. (2010). Understanding the Use of Elicitation Approaches for Effective Requirements Gathering. Teoksessa J. Hall., H. Kandl, L.

Lavazza, G. Buchgeher & O. Takaki (toim.), Proceedings of the 5th International Conference on Software Engineering Advances (ICSEA). Nice, France, 22–27 August. (s. 325–330).

Chow, T., Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems & Software, Agile Product Line Engineering*, 81(6), pp. 961–971.

Collier, K. (2011) *Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing*, Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Indiana: Addison-Wesley Professional.

Dalpiaz, F., Brinkkemper, S. (2018) *Agile Requirements Engineering with User Stories*. IEEE 26th International Requirements Engineering Conference.

Darwish, N., Megahed, S. (2016) *Requirements Engineering in Scrum Framework*. International Journal of Computer Applications (0975 – 8887) Vol. 149(8)

Digia. (2019a) Digia yrityksenä. Viitattu 28.8.2019, osoitteesta: <https://digia.com/yritys/>

Digia. (2019b) Palvelumme. Viitattu 28.8.2019, osoitteesta: <https://digia.com/palvelumme/>

Düchting, M., Zimmermann, D., Nebe, K. (2007). Incorporating user centered requirement engineering into agile software development. In Jacko, J. A., editor, *Human-Computer Interaction. Interaction Design and Usability*, 12th International Conference, HCI International 2007, Beijing, China, July 22-27, 2007, Proceedings, Part I, volume 4550 of *Lecture Notes in Computer Science*, pages 58–67. Springer.

Dydå, T., Dingsoyr, T. (2008) Empirical studies of agile software development: A systematic review. *Information and Software Technology*. pp. 833.859.

Eberlein, A., Maurer, F. (2003) *Requirements Engineering and Agile Software Development*, University of Calgary.

Elshandidy, H., Mazen, S., 2013, Agile and Traditional Requirements Engineering: A Survey, *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 9.

Eriksson, P., Kovalainen, A. (2015). Qualitative methods in business research: A practical guide to social research. Sage.

Fowler, M., Highsmith, J. (2001), "The Agile Manifesto", Software Development Magazine, August.

Garret, J. (2011) The Elements of User Experience. User-centered design for the web and beyond. Pearson Education.

Girvan, L., Paul, D. (2017;2016). Agile and business analysis: Practical guidance for IT professionals (1st ed.). London: BCS Learning & Development Ltd.

Guba, E. G., Guba, Y. A. L., Lincoln, Y. S. (1994) Competing Paradigms in Qualitative Research, Handbook of qualitative research. doi: <http://www.uncg.edu/hdf/facultystaff/Tudge/Guba%20&%20Lincoln%201994.pdf>.

Gulliksen, J., Göransson, B., Boivie, I., Bolomkvist, S., Persson, J., Cajander, Å. (2003) Key principles for user-centered system design. Behavior & Information Technology, 22(6), pp. 397-409.

Göransson, B. (2004), User-centered system design. Designing usable interactive systems in practice. Comprehensive Summaries of Uppsala Dissertations from Faculty of Science and Technology 981.

Haikala, I., Mikkonen, T. (2011), Ohjelmistotuotannon käytännöt. Talentum. Helsinki.

Heinilä, J. (2005) User-Centred Design: Guidelines for Methods And Tools.

Hull, E., Jackson, K., Dick, J. (2005), Requirements Engineering: Second Edition. Tunbridge Wells, England: Gray Publishing.

IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers. IEEE Standards Board.

ISO 13407, (1999), Human-Centred Design Processes for Interactive Systems. Viitattu 17.7.2019, osoitteesta: <https://www.iso.org/standard/21197.html>

ISO (2010) ISO 9241-210: Ergonomics of human–system interaction - Part 210: Human-centred design for interactive systems.

JUHTA. 2009. JHS 173 ICT-palvelujen kehittäminen: Vaatimusmäärittely. Versio 1.1.

Kasanen, E., Lukka, K., Siitonen, A. 1993. The Constructive Approach in Management Accounting Research. *Journal of Management Accounting Research*. pp. 243-264.

Kotaiah, B., Khalil, M. A. (2017). Approaches for development of Software Projects: Agile methodology. *International Journal of Advanced Research in Computer Science*; Udaipur, 8(1).

Kotonya, G., Sommerville, I. (2002). *Requirements Engineering Processes and Techniques*. John Wiley & Sons. Great Britain.

Lamsweerde van, A. (2009). *Requirements Engineering From System Goals to UML Models to Software Specifications*. Wiley, NJ, USA.

Law, E., Roto, V., Hassenzahl, M., Vermeeren, A., Kort, J. (2009) Understanding, Scoping and Defining User eXperience: A Survey Approach. *SIGCHI Conference on human factors in computing systems*. ACM. pp. 719 – 728.

Leffingwell, D. (2011). *Agile software requirements: lean requirements practices for teams, programs, & the enterprise*. Agile software development series. Addison-Wesley.

Lucassen, G., Dalpiaz, F., Werf, J., Brinkkemper, S. (2015) Improving agile requirements: the Quality User Story framework and tool. *Requirements Eng* (2016) pp. 383–403.

Lucia, A., Qusef, A. (2010) Requirement Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence*, vol. 2(3).

Mao, J., Vredenburg, K., Smith, P. W., and Carey, T. (2005). The state of user-centered design practice. *Commun. ACM*, 48(3) pp.105-109.

Misra, S. (2011) Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*. Vol 29. pp. 972-980.

Nakao, Y., Moriguchi, M., Noda, H. (2014) Using agile software development methods to support human-centered design. *NEC Technical Journal*, 8(3), pp. 37-40.

Nielsen Norman Group (2009) User experience: Our definition. Nielsen Normal Group. Viitattu 16.7.2019, osoitteesta: <https://www.nngroup.com/articles/definition-user-experience/>

Nielsen, J. (1993) *Usability Engineering*. San Francisco, CA: Morgan Kaufmann.

Norman, D. (2013) *The Design of Everyday Things*. Revised & Expanded Edition. New York: Bsic Books.

Nuseibeh, B., Eastbrook, S. (2000). *Requirements Engineering: A Roadmap*. Teoksessa *Proceedings of the Conference on the Future of Software Engineering*, Limerick, Ireland, June 4–11. (s. 35–46). New York: ACM.

Ojasalo K., Moilanen T. & Ritalahti J. 2009. *Kehittämistyön menetelmät. Uudenlaista osaamista kehittämistoimintaan*. Helsinki: WSOYpro.

Oyegoke, A. (2011) The constructive research approach in project management research. *International Journal of Managing Projects in Business*. Vol. 4, Iss. 4 pp. 573-595.

Paetsch, F., Eberlein, A., Maurer, F. (2003) Requirements engineering and agile software development. in *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 308–313.

Pandey, D., Suman, U., Ramani, A. (2010) An Effective Requirement Engineering Process Model for Software Development and Requirements Management. in *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 287–291.

Parantainen, J. (2007) Rakenna palvelusta tuote 10 päivässä. Tuotteistaminen. Talentum media oy.

Pavelin, K., Pundir, S., Cham, J. (2014) Ten simple rules for running interactive workshops. PLoS Comput Biol 10(2).

Pohl, K., Rupp, C. (2011), Requirements Engineering Fundamentals. A Study Guide for the Certified Professional for Requirements Engineering Exam. Foundation Level IREP compliant. Rocky Nook Inc. Santa Barbara.

Pollack, J. (2007) The changing paradigms of project management. International Journal of Project Management. Vol. 25, pp. 266-274.

Qu, S.Q., Dumay, J. (2011). The qualitative research interview, Qualitative Research in Accounting & Management, Vol. 8(3), pp. 238-264.

Quesf, A., (2010), Requirements Engineering in Agile Software Development, Journal of Emerging Technologies in Web Intelligence, Vol.2, No.3.

Ramesh, B., Cao, L., Baskerville, R. (2010) Agile requirements engineering practices and challenges: an empirical study. Journal compilation. Blackwell Publishing Ltd.

Reason, B., Løvlie, L., & Brand, FM. (2015). Service Design for Business : A Practical Guide to Optimizing the Customer Experience, John Wiley & Sons, Incorporated, Hoboken. Available from: ProQuest Ebook Central.

Rubin, J., Chisnell, D. (2008) The art of agile development (2nd ed.) Sebastopol, Canada: O'Reilly Media Inc.

Robertson, S., Robertson, J. (2011), Mastering the Requirements Process. AddisonWesley. Upper Saddle River (NJ).

Saunders, M. N. K., Lewis, P., Thornhill, A. (2009). Research methods for business students (5th ed.). New York: Prentice Hall.

Schwaber, K., Beedle, M. (2001). Agile software development with scrum. Upper Saddle River: Prentice-Hall.

Schwaber, K., Sutherland, J. (2017) The Scrum Guide. The Definition Guide to Scrum: The Rules of the Game. Viitattu 18.6.2019, osoitteesta: <https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide>

Schwaber, K. (1997). SCRUM Development Process. Teoksessa J. Sutherland, C. Casanave, J. Miller, s. Patel, & G. Hollowell (Toim.), Business Object Design & Implementation (ss. 117–134). London: Springer London. Viitattu 23.6.2019, osoitteesta http://link.springer.com/10.1007/978-1-4471-0947-1_11

Schön, E., Thomaschewski, J., Escalona, M. (2017) Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces 49. pp. 79-91.

Schön, E., Thomaschewski, J., Escalona, M. (2017) Identifying Agile Requirements Engineering Patterns in Industry. Computer Standards & Interfaces.

Sommerville, I. (2010) Software Engineering, 9th edition, Nine. New York : Addison-Wesley.

Stickdorn, M., Hormess, M.E., Lawrence, A., Schneider, J. (2018). This Is Service Design Doing: Applying Service Design Thinking in the Real World, 1st edn, O'Reilly Media, Sebastopol.

Stickdorn, M. ja Schneider, J. (2011) This is service design thinking. 7th p, BIS Publishers. 7th s. Amsterdamn: BIS Publishers.

Sutcliffe, A. (2009) The Encyclopedia of Human-Computer Interaction. 2nd edition Requirements engineering. Interaction design foundation.

Teixeira, J., Patrício, L., Nunes, N. J., Nóbrega, L., Fisk, R. P., Constantine, L. (2012). Customer experience modeling: From customer experience to service design. Journal of Service Management, 23(3), 362-376.

Thomas, E. Magilvy, J. K. (2011) Qualitative Rigor or Research Validity in Qualitative Research, Journal for Specialists in Pediatric Nursing.

Toikko T. & Rantanen T. 2009. Tutkimuksellinen kehittämistoiminta. Tampere: Tampere University Press.

VTT. (2019) User-Centred Design. Guidelines for Methods and Tools. VTT Information Technology. University of Oulu. Dept. of Information processing science. Philips Research. Philips Applied Technologies.

Vähäniitty, J. (2012) Towards Agile Product and Portfolio Management. Doctoral Dissertations 15/2012. Aalto University publication series.

Walton, D. (2014). Abductive Reasoning (eBook edition.). Tuscaloosa, UNITED STATES: University of Alabama Press.

Whittemore, R., Chase, S. K., Mandle, C. L. (2001) Validity in qualitative research, Qualitative Health Research.

Yin, R. K. (2013). Validity and generalization in future case study evaluations. Evaluation, 19(3), 321-332.

Ylikangas, M., Kauppinen, M. (2016) Requirements Engineering – From the Perspective of Customer and Users. Software design and modelling.

Young, R.R. (2004) The Requirements Engineering Handbook. Norwood, United States of America: Artech House, Inc.

LIITTEET

LIITE 1: Teorian pohjalta muodostettu viitekehys (Salattu)

LIITE 2: Haastattelurunko

LIITE 3: Teorian ja empirian yhdistämä viitekehys (Salattu)

LIITE 4: Analyysivaiheen rikastettu viitekehys (Salattu)

LIITE 2: Haastattelurunko

Haastattelurunko

1. Mikä on haastateltavan nimi ja työnkuva?
2. Kuinka pitkään olet työskennellyt tehtävissäsi?
3. Oletko ollut kuinka pitkään mukana vaatimusmäärittelyissä?
4. Miten toiminnallisten vaatimusten määrittäminen on tähän asti yrityksessä toteutettu?
 - a. Onko jotain yleisesti sovittuja käytänteitä? Mallia, jota noudatetaan?
 - b. Keitä prosessiin osallistuu?
 - c. Miten asioita on dokumentoitu? Millaisia dokumentteja on muodostettu?
Onko tähän jotain sovittua käytäntöä?
 - d. Mikä on ollut erityisen onnistunutta?
 - e. Millaisia ongelmia on kohdattu?
 - a. Mitä mieltä olet, miten vaatimusmäärittely tulisi tehdä?
5. Esitellään kirjallisuuskatsauksen kautta koostettu malli
6. Käydään läpi millaisia ajatuksia malli herättää
 - a. Mitä parannettavaa
 - b. Mitkä asiat hyviä
 - c. Vapaata kommentointi
7. Mitkä ovat syitä miksi mallin ottaisi käyttöön tai ei ottaisi käyttöön?